



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación :

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

Título de proyecto:

**DESARROLLO DE APLICACIÓN WEB 2.0
PARA LA CREACIÓN, COMPOSICIÓN Y
EDICIÓN DE IMÁGENES COLABORATIVA.**

Alumno: Kudinov Oleh

Tutor: Oscar Ardaiz Villanueva

Pamplona, 25 de Febrero de 2010

1. INTRODUCCIÓN.....	3
2. ANTECEDENTES Y OBJETIVOS.....	4
2.1. Antecedentes.....	4
2.1.1 Picnik.com.....	4
2.2 Objetivos.....	6
3. DESAROLLO DEL TRABAJO.....	8
3.1 EDITOR ONLINE.....	9
3.1.1 ANÁLISIS.....	9
3.1.1.1 ANÁLISIS DE LOS REQUISITOS.....	9
Análisis y especificación de los requisitos.....	9
Búsqueda de información.....	10
3.1.1.2 ANÁLISIS DEL SISTEMA.....	10
Casos de uso.....	10
Diagramas de clases.....	16
Separación por módulos.....	19
Diagrama Entidad-Relación.....	21
3.1.2 DISEÑO E IMPLEMENTACIÓN.....	23
Paso a tablas del diagrama ER.....	23
Elección de la tecnología a usar.....	27

Módulos.....	29
Acceso.....	29
Administración.....	30
Gestión de imágenes.....	30
Gestión de espacios.....	34
Edición.....	36
4. ANÁLISIS DE RESULTADOS.....	50
5. COSTE Y PLANIFICACIÓN.....	52
5.1 PLANIFICACIÓN.....	52
5.2 COSTES.....	52
6. CONCLUSIONES Y PASOS FUTUROS.....	53
6.1 CONCLUSIONES.....	53
6.2 PASOS FUTUROS.....	55
7. MANUAL DE ADMINISTRADOR.....	56
8. MANUAL DE USUARIO.....	58
9. BIBLIOGRAFÍA.....	66

1. INTRODUCCIÓN

Hoy en día las imágenes se crean, editan e integran con paquetes como Photoshop o GIMP. Sin embargo, muchas veces sólo un número de base de dicha funcionalidad es necesario por parte de cualquier usuario. Aplicaciones de edición de imágenes en línea están apareciendo en los sitios como www.flickr.com o www.dumprr.net. Los editores online tienden a ofrecer cada vez más cantidad de funcionalidades de las que tienen las aplicaciones como Photoshop, GIMP, Pixelmator. Pero aún les queda un largo camino para poder ofrecer las mismas funcionalidades que sus parientes.

A muchos les debe de haber pasado que al no contar con un editor de imágenes en su computadora no pudieron darle un retoque a sus imágenes. Así pues, ahora con esta aplicación necesitará solamente un navegador web.

Los editores de imágenes online cuentan con varias ventajas, entre las que se pueden destacar:

- No es necesario instalar ningún software.
- En su mayoría cuentan con las funciones más utilizadas.
- La velocidad de respuesta es muy buena, nada de demoras de varios minutos para aplicar un efecto.
- Podemos extraer las imágenes directamente desde el sitio web sin necesidad de descargarlas para luego subirlas.

Los editores online como flickr.com son aplicaciones muy útiles. Sin embargo no ofrecen la funcionalidad de colaboración para la edición simultánea de las imágenes y además su editor está desarrollado con flash, lo que significa que hay que descargar antes un plugin para su navegador. También hay que considerar que a los usuarios de flickr.com les resulta complicado comprender todas las opciones que tiene esa aplicación y por eso existen diferentes manuales online.

El proyecto consistirá en el desarrollo de una aplicación que ofrece a los usuarios las funciones básicas de edición y permite realizarlo de forma colaborativa para la edición simultánea de las imágenes.

2. ANTECEDENTES Y OBJETIVOS

2.1 Antecedentes

Para la realización de este proyecto se analizaron muchas aplicaciones existentes en la actualidad similares a la aplicación que se debe desarrollar para saber como funcionan y que ofrecen. Después se escogió una para hacer un estudio más detallado y sacar de ella los conceptos básicos. Estos se describen a continuación.

2.1.1 Picnik.com

Picnik.com es un editor de imágenes online, tiene como características principales la facilidad de uso y sobre todo la integración perfecta con otros servicios web 2.0 como Facebook, Picasa Web o Flickr.

Con este potente editor web de Picnik.com puedes modificar y transformar tus imágenes digitales favoritas sin necesidad de instalar un programa que ocupe mucho espacio en el ordenador. Así, en tan sólo unos cuantos pasos, es posible variar el tamaño, las tonalidades o el enfoque de cualquier captura de forma totalmente gratuito.

Además, su interfaz está completamente traducida al español. Aunque podemos trabajar con Picnik.com sin darnos de alta, es conveniente hacerlo para que el programa reconozca nuestro nuevo perfil de usuario.

A la hora de editar, podemos subir nuestra imagen utilizando un gestor de ficheros para localizar el archivo gráfico deseado dentro de los discos del PC. Además, podemos cargar las imágenes instantáneas desde otros portales especializados, como Flickr, Picasa, Facebook, Photobucket y Webshots. Hecho esto, las fotos almacenadas en flash son trasladadas al instante a la base de datos de Picnik.com. Otra forma de conseguir capturas interesantes para su edición consiste en utilizar la Red, ya sea a través de un vínculo directo, la especificación URL de un sitio web o con la búsqueda de un buscador de fotos.

En menú de edición se puede observar que su barra de funciones incluye una gran variedad de utilidades. En primer lugar, corregir automáticamente aplicando un filtro instantáneo que elimina todos los valores demasiado acentuados que contenga el fichero, ya sea en relación a la intensidad del color, su contraste o su brillo. A ésta le sigue la opción de Rotar, capaz de girar la imagen en todas las direcciones. Igualmente útil, Recortar define una gran variedad de proporciones y tamaños, con formatos tan variados como una foto de Blogger, una miniatura de Youtube o un fondo de escritorio de 1280 x 800, entre otros.

Para ajustar la imagen a cualquier entorno, existe la utilidad cambiar tamaño. Gracias a ella, puedes dimensionar la captura, manteniendo sus proporciones originales. Por otro lado, la herramienta exposición, mejora el contraste entre las áreas que contienen píxeles de diferentes tonalidades.

Después del color esta una opción que regula los contornos de las figuras incluidas en la imagen, ganando una mayor nitidez. Y para finalizar el apartado de edición, puedes utilizar ojos rojos para corregir esa deficiencia que aparece a veces en algunos retratos. Dentro de la pestaña crear encontraremos multitud de efectos divertidos.

Toda la aplicación como casi todos los editores online están desarrollados en flash , así podemos ver que toda la página entera se carga, avisando que debemos esperar un rato. Así antes de usar, se tiene que descargarse el programa entero de flash, por lo que habría que esperar, pero la ventaja es que es mas fácil implementarlo en flash. En este Proyecto Fin de Carrera de 3er curso no se usara flash.

2.2 Objetivos

El objetivo de este proyecto es construir una aplicación, empleando la tecnología Web 2.0, la cual permitiese a los usuarios el intercambio de sus trabajos de desarrollo o de espacios de trabajo de imágenes a los que vamos a llamar espacios de trabajo, aumentando de esta manera la creatividad y ofreciendo a los usuarios la posibilidad de acabar o ayudar en edición de imágenes de trabajos de los otros usuarios.

Los usuarios notarán la mayor facilidad de uso y aparición de herramienta a los sitios que de normal suele usar y navegar, por tanto se usará la tecnología Web 2.0 para poder arrastrar las imágenes.

La aplicación deberá formar una parte de sitio web como facebook o paginas que son galerías de imágenes, y cualquier usuario que está observando las imágenes en alguna de las categorías, tenga una posibilidad de editar las o subir sus propias imágenes para su posterior edición. Así también, guardar los trabajos de edición que están en desarrollo y pueden ser seguidamente desarrollados por otros usuarios.

El usuario o administrador será capaz de elegir y buscar categoría y también añadir cualquier imagen en categoría que quiera. Por otra parte tendrán la posibilidad de visualizar, añadir y editar todas las imágenes de que se dispone en cada categoría. Además, podrán realizar búsquedas básicas de las imágenes que les interesan. También podrán trabajar con los espacios de trabajo de los demás. Con los espacios de trabajo, los usuarios podrán visualizar, crear, editar, realizar búsquedas y guardar su espacio de trabajo en el sitio de espacios de trabajo compartido. A los administradores les estará permitido borrar cualquier imagen o espacio de trabajo y a los usuarios solo imágenes o espacios de trabajo que están guardados por ellos mismos.

Los usuarios dispondrán de una categoría de imágenes especial. Es de fotos privados, son los fotos subidos por un usuario y a los que no podrán ver los demás.

Para trabajar con editor se implementaran las operaciones con los que usuarios podrán realizar distintas operaciones con las imágenes como redimensionar, dibujar con diferentes colores, borrar, ampliar o reducir el espacio de trabajo(es cuando el menú con las operaciones y todas las imágenes se cambian de tamaño), rotar con cualquier ángulo, fusionar todas las imágenes, recortar, hacer el efecto de espejo, hacer copias de imágenes, crear hojas en blanco y guardar su trabajo en el sitio de espacios de trabajo compartido. Y todo esto tendrá que funcionar con muchas imágenes a la vez moviendo las y situando las una encima de otra.

El editor tendrá un menú que se podrá mover por cualquier lugar de espacio de edición. Las dimensiones de espacio de trabajo serán muy grandes para así poder trabajar con las imágenes de cualquier tamaño. Cada funcionalidad del menú de edición tendrá un dibujo asociado para que cualquier usuario que no sepa español pueda trabajar fácilmente con el editor.

En caso si una imagen será de tamaño grande, habrá que controlar las operaciones de usuario de manera que no podrá hacer nada hasta que la carga no haya finalizado, para que el usuario sepa cuanto tiempo le resta para finalizar la descarga de una imagen después de cada operación.

Como aplicación es por internet, entonces habrá que reducir al máximo el tiempo de espera de usuario. Así pues, tendremos que utilizar herramientas que nos permiten mandar datos de servidor a cliente sin recarga previa de la pagina.

3. DESARROLLO DEL TRABAJO

Para llevar a cabo nuestro proyecto, el proceso de desarrollo se ha dividido en las etapas de análisis, diseño, codificación y pruebas de la aplicación. Así pues se explicaran los diferentes pasos realizados para cumplir los objetivos predeterminados. Como GD library y el editor ImageMagick proporcionan los APIs de sus funciones, nuestra aplicación usará estas funciones. Gracias a esto nuestro proyecto se simplificará mucho y nuestra aplicación final será de menos tamaño.

GD es una librería de código fuente abierto para la creación dinámica de imágenes por los programadores. GD está escrito en C, y los "contenedores" están disponibles para Perl, PHP y otros lenguajes. GD crea PNG, JPEG y GIF, y otros formatos. GD es comúnmente utilizado para generar tablas, gráficos, imágenes en miniatura.

ImageMagick es un editor de imágenes. Puede leer, convertir y escribir imágenes en una variedad de formatos, incluyendo DPX, EXR, GIF, JPEG. Así pues, nuestra aplicación será encargada de interactuar con el usuario final. El cliente no usará ni conocerá todas las posibilidades de GD library y ImageMagick, ver la figura 1.

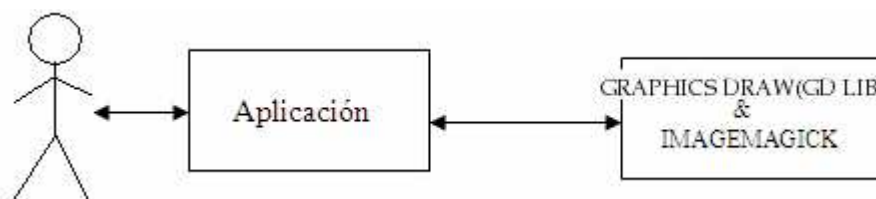


Fig.1

Es importante ver la arquitectura que se usará. Vamos a suponer que nuestra aplicación tendrá la base de datos, porque es muy común en aplicaciones de este tipo. La arquitectura será de tres capas y tenemos que añadir una capa más para el programa de cliente. La arquitectura resultante es la figura 2.

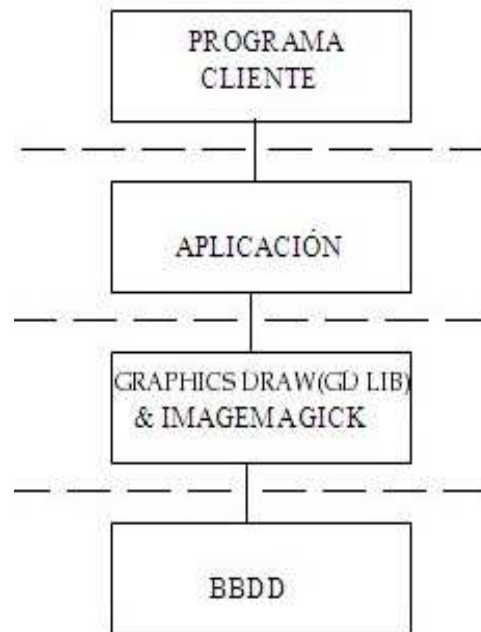


Fig.2

3.1 EDITOR ONLINE

3.1.1 ANÁLISIS

Este paso se divide en dos pasos: análisis de los requisitos y análisis del sistema.

3.1.1.1. ANÁLISIS DE LOS REQUISITOS

El objetivo de este paso es establecer la base teórica muy clara para no tener que decidir los aspectos importantes durante diseño. Eso significa que hay que obtener los requisitos de sistema, buscar la información importante para el proyecto. Como los dos apartados son dependientes entre si los dos pasos se realizan uno detrás de otro.

Análisis y especificación de requisitos

En este paso tenemos que conseguir los datos explicados de punto 2.1 Objetivos de este documento. La información es obtenida en las reuniones con el tutor y otros medios como análisis de las aplicaciones similares. El objetivo de proyecto fue determinado cuando el alumno y tutor eran de acuerdo.

Es muy importante establecer todos los requisitos y que estos no cambian. En caso si los requisitos están mal especificados esto nos puede llevar a finalización de proyecto sin haber conseguido los objetivos, por este motivo este paso tiene gran importancia en el análisis de nuestra aplicación.

Búsqueda de información

El objetivo es buscar información que nos puede servir de ayuda durante todo el desarrollo de aplicación. Se analizaron distintos enlaces Web, los cuales se especifican en bibliografía de este documento. La información buscada fue de siguiente manera

1. Buscar aplicaciones o sistemas similares
2. Buscar tecnologías posibles de ser utilizadas
3. Manuales sobre lenguaje de programación

3.1.1.2 ANÁLISIS DEL SISTEMA

Cuando ya sabemos los objetivos de aplicación, ya podemos realizar una visión clara de problema global de aplicación. Esto nos ayudara diseñar de forma más eficiente.

Casos de uso

Para hacer los casos de uso de sistema hay que primero determinar los autores y averiguar los requisitos. En principio tenemos claro que hay un administrador de nuestra aplicación. También existe usuario registrado del sistema, que puede realizar prácticamente todas las funciones disponibles. También hay usuarios no registrados, que tienen solo algunas funciones disponibles. En resumen tendremos siguientes autores: usuario no registrado, usuario registrado y administrador.

Las funcionalidades requeridas por el sistema se pueden dividirse en tres grupos: funciones básicas, funciones de editor, funciones de sitio compartido de espacios de trabajo, funciones de gestión de categorías de imágenes y funciones especiales.

Así pues, el diagrama resultante es la figura 3.

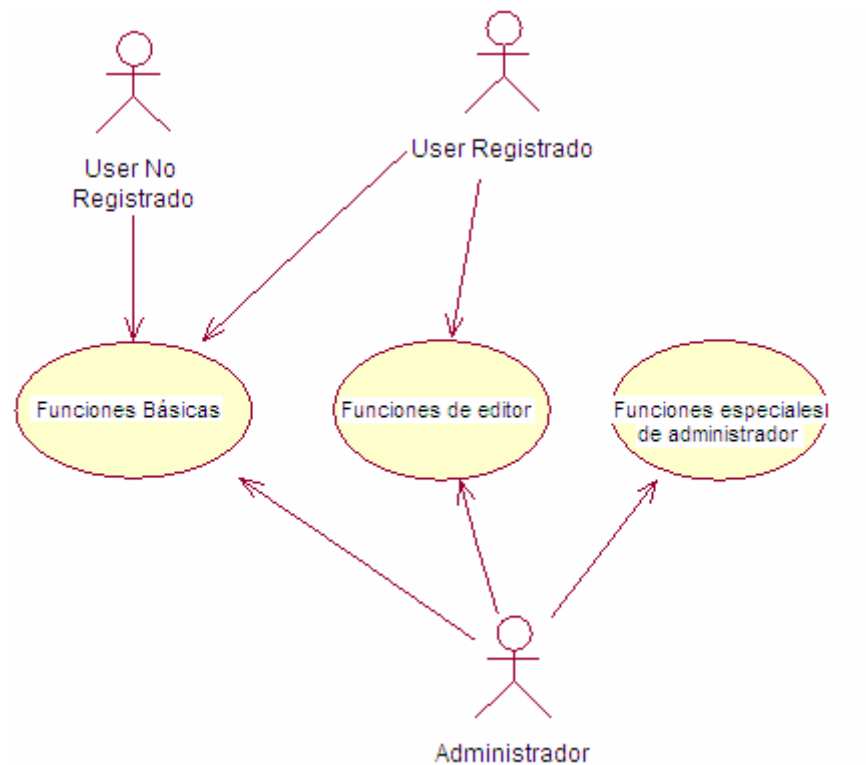


Fig.3

Ahora se realizará la descomposición de cada caso de uso. Se describirán los casos de uso para cada autor.

Casos de uso de funciones básicas.

Registro: cada usuario podrá crear una cuenta en el sistema. Para registrarse el usuario deberá introducir los datos como el nombre de usuario y la contraseña. Si el usuario existe o la contraseña está mal el sistema le avisara y el usuario intentara de nuevo hasta conseguir un registro valido.

Login: El login es autenticación en momento de ingresar al sistema. Para loguearse se le pedirá el nombre y la contraseña. Si la cuenta con este nombre existe y la contraseña está bien entonces pasara a ser el usuario registrado sino le ofrecemos un enlace para realizar el registro.

Logout: El usuario registrado tendrá una opción de cerrar su sesión en el sistema. Para esto en cada ventana tendrá un enlace con esa opción.

Visualizar imágenes: Como las imágenes en cada categoría van a ser de tamaño pequeño, tendremos que agrandarlas cada vez que el usuario pasa con el ratón sobre ellas. También, el usuario tendrá que saber el nombre de cada imagen que está viendo y esta información también le va a aparecer en una ventanilla muy pequeña sobre la imagen agrandada.

Visualizar los espacios de trabajo: En el sitio compartido de espacios de trabajo, tendremos la posibilidad de ver todos los espacios de trabajo de la misma forma que las imágenes, agrandando los y viendo que imágenes hay en cada espacio. También nos aparecerá la información de cada espacio de trabajo como: nombre de usuario , nombre de espacio , el día cuando este espacio fue añadido.

Añadir imágenes: El usuario registrado o no registrado podrá entrar en cualquier categoría y subir su imagen. También lo puede hacer el administrador.

Borrar imágenes: Cada usuario registrado podrá borrar sus imágenes que antes había subido, pero no podrá borrar nada de los demás. El administrador podrá borrar todo.

Borrar espacios de trabajo: Todos menos los no registrados, podrán borrar sus espacios de trabajo que están en el sitio compartido, pero no podrán borrar nada de los demás. El administrador será capaz de borrar todo lo que quiera.

Elegir categoría: Para elegir una categoría los usuarios dispondrán de una lista ordenada con los nombres de categorías. También lo podrán hacer mediante una bola giratoria.

Buscar imágenes: Todos podrán hacer una búsqueda por el nombre de imagen. Si alguna categoría que tenemos coincide con la palabra buscada entonces en el resultado de búsqueda saldrán todas las imágenes de esta categoría.

Buscar espacios de trabajo: Todos podrán hacer una búsqueda dentro del sitio compartido de espacios de trabajo. La búsqueda va ser personalizada. Podremos cambiar el modo de búsqueda y buscar por el nombre de espacio, o por la fecha o por el nombre de usuario.

Visualizar contenido de la página. Usaremos las hojas de estilos para que el usuario pueda disfrutar de una buena estructura de página.

Gestionar imágenes: Dispondremos de posibilidad de arrastrar las imágenes. Incluye el borrado y la subida de las imágenes.

Pasar imagen a editor: Guarda la imagen en el espacio de trabajo de usuario actual y inicia el editor, para iniciar la edición de esta imagen.

Así pues, el diagrama resultante es la figura 4.

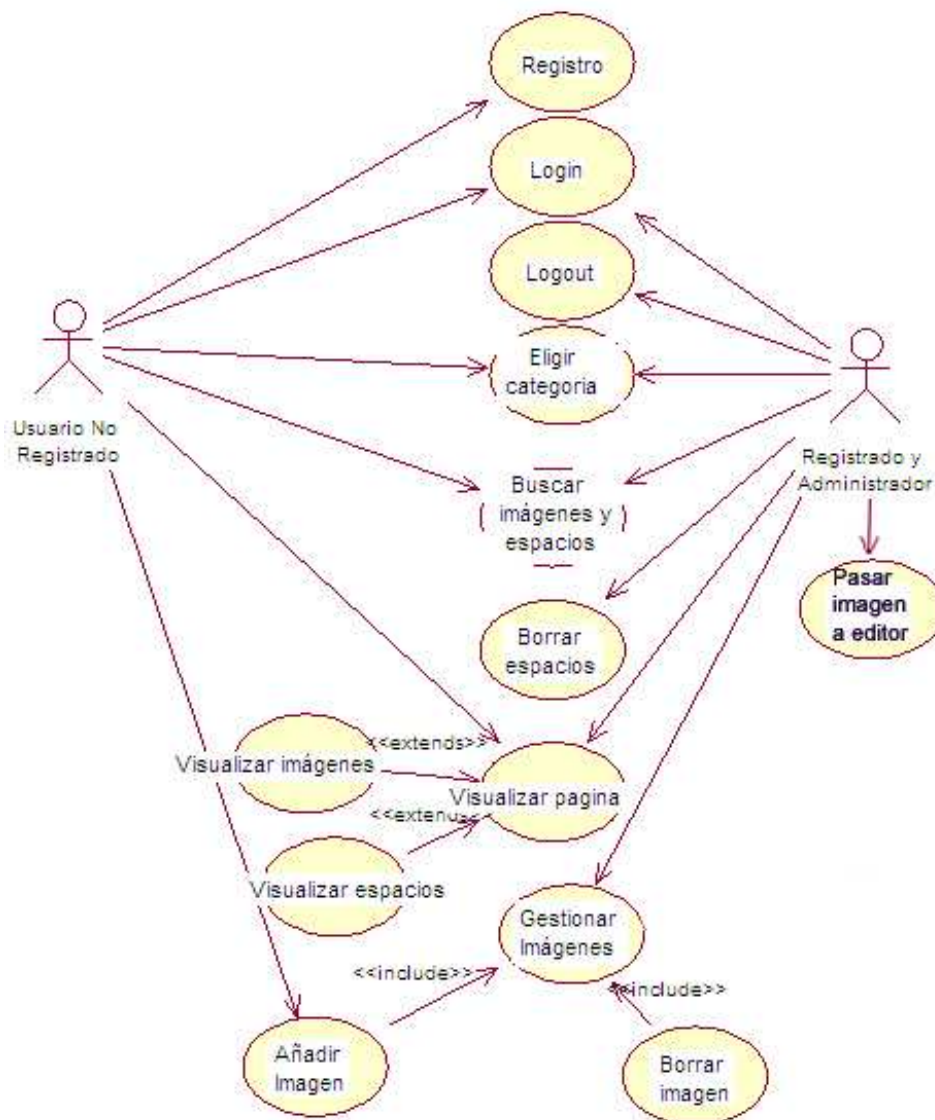


Fig.4

Casos de uso de funciones especiales de administrador

En cualquier página debe existir un administrador que tiene que gestionar la página web. Así pues, en nuestro sistema el administrador podrá administrar a los usuarios y el contenido de la página. Esto sirve en casos si las imágenes subidas son de contenido inadecuado, para poder eliminarlos. Tenemos los siguientes casos de uso:

Borrar usuarios: el espacios de trabajo del mismo también se eliminara, y todos los contenidos de este usuario pasarán a ser de administrador.

Borrar imágenes o espacios de trabajo: el administrador podrá borrar cualquier imagen o espacio de trabajo independiente de quien es el propietario del mismo.

Modificar imágenes o espacios de trabajo: el administrador podrá abrir cualquier imagen, editarla y volver a subirla.

El diagrama de casos de uso es la figura 5.

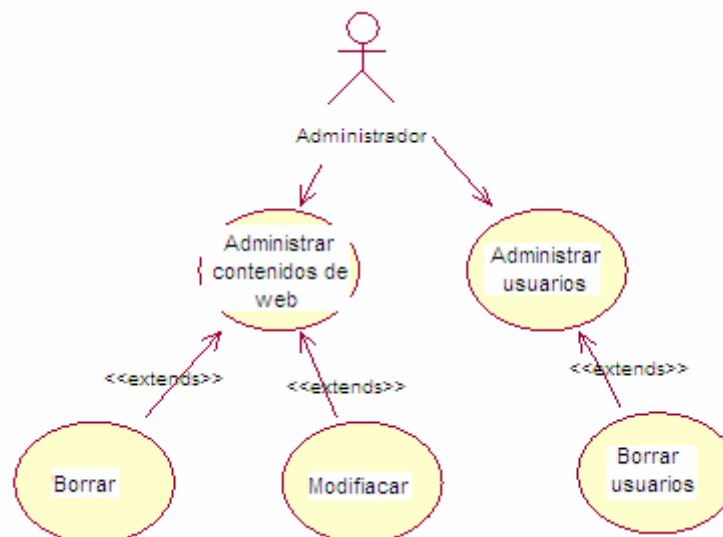


Fig.5

Casos de uso de funciones de editor

Los actores que podrán trabajar con el editor son el usuario registrado y el administrador. Pero estos dos actores dispondrán de los mismos casos de uso.

Los actores tendrán los siguientes casos de uso:

Redimensionar: El usuario podrá redimensionar una imagen de manera más cómoda posible. Así apuntaría el cursor del ratón en una de las esquinas y le aparecerá un aviso que puede empezar a arrastrar, después de esto comenzaría a arrastrar y el tamaño se cambiaría, enseñando en cada momento las dimensiones actuales del imagen.

Cambiar zoom: El usuario podrá aumentar o disminuir el zoom de su espacio de trabajo. En este caso todas las cosas del mismo espacio se cambiaran de tamaño.

Dibujar: Seleccionando una de las imágenes, el usuario será capaz de dibujar con los diferentes colores y tamaños.

Cambiar color: Para poder dibujar con diferentes colores habría que seleccionar antes algún color. Con esta funcionalidad los actores podrán cambiar el color de manera muy sencilla.

Cambiar tamaño de pincel. Para poder dibujar con diferentes tamaños el usuario dispondrá de diferentes tamaños de pincel a elegir.

Borrar con pincel: Este caso de uso es parecido al caso de uso de dibujar, pero aquí el pincel pinta con el color blanco y el tamaño es más grande

Eliminar imagen: Los actores podrán borrar cualquier imagen arrastrándola a la papelera.

Rotar: Este caso de uso permite a los actores cambiar el ángulo de cualquier imagen de espacio de trabajo.

Fusionar: Los actores podrán fusionar todas las imágenes que tienen en su espacio de trabajo.

Guardar: Los actores tendrán la posibilidad de compartir sus trabajos en cualquier estado de desarrollo. Esto significa que podrán guardar sus espacios de trabajo en el sitio compartido de espacios de trabajo. Para poder enseñar en el espacio compartido el estado de trabajo, todas las imágenes del mismo deben ser fusionadas, por lo tanto este caso de uso usará el caso de uso fusionar.

Reemplazar espacio de trabajo: El usuario podrá reemplazar su espacio de trabajo por cualquier otro que está en el sitio compartido.

Recortar: Con este caso de uso podremos recortar cualquier área cuadrada de imagen seleccionada.

Subir imagen: El usuario podrá subir cualquier imagen desde su PC o desde cualquier categoría a su espacio de trabajo para su posterior edición.

Copiar imagen: El usuario será capaz de hacer copias de las imágenes que selecciona.

Crear imagen en blanco: Este caso de uso será necesario para crear imágenes en blanco.

El diagrama de casos de uso es la figura 6.

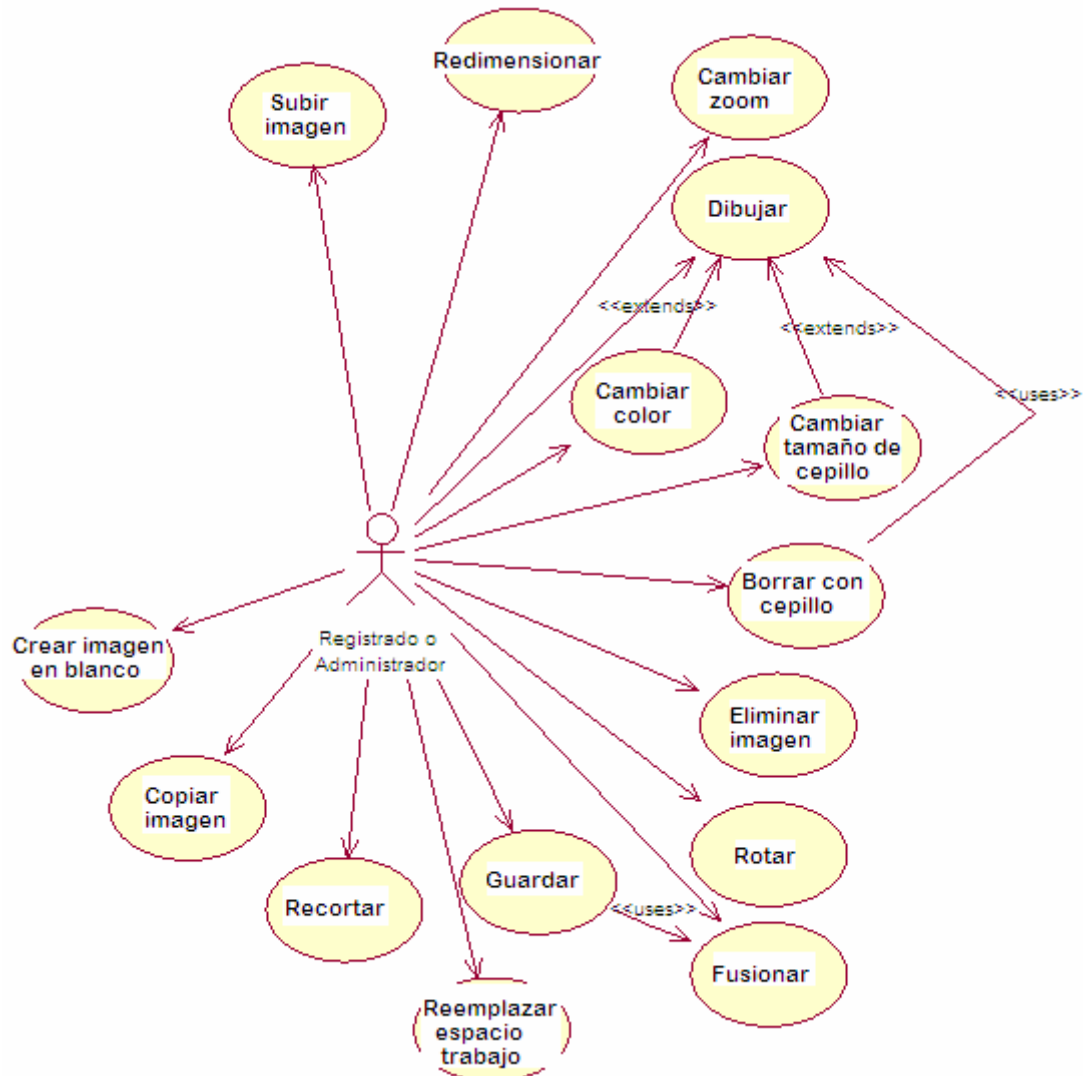


Fig.6

Diagrama de clases

Como la aplicación estará implementada con javascript, ajax, php y html, solo podemos realizar el diagrama de javascript y ajax, porque en php no haremos objetos. En principio haremos diagrama de clases relacionadas con funciones básicas. Las funciones y atributos que tienen menor importancia se omiten, pero se describirán posteriormente. Además todas las funciones que están en php también se describirán en siguientes apartados.

Diagrama de clase de funciones básicas

Cada página en javascript es un objeto de window así pues, vamos a tener las tres ventanas para estas funcionalidades, son: galería, sitio compartido de espacios de trabajo y una ventana más para hacer login. Galería es ventana que contiene todas las categorías con sus imágenes y esta clase es galería. Luego tenemos una ventana de espacios de trabajo almacenados, esta es la clase

EspaciosTrabajo. La ventana de de galería tendrá un objeto SWFObject. Es una bola giratoria y sirve para seleccionar una de las categorías. La ventana de galería también contendrá la ventana de edición en el diagrama llamada Edit, es porque tenemos que abrir el editor desde la ventana de galería. Lo mismo pasa con la ventana de espacios de trabajo almacenados.

La ventana de inscripción de usuario no tiene ninguna función y ningún atributo en javascript y solamente esta implementado en html y php, por eso se omite en esa diagrama.

El diagrama resultante es la figura 7.

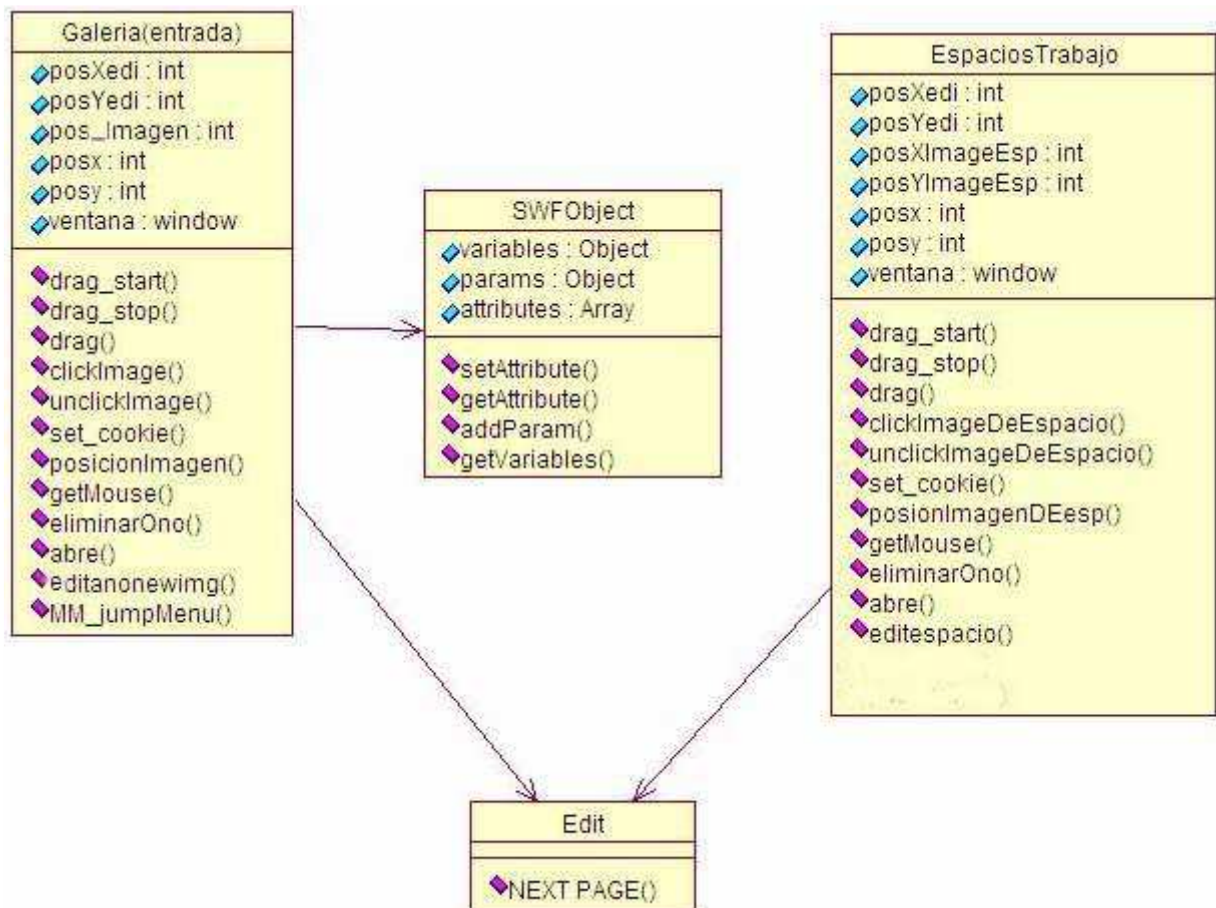


Fig.7

Diagrama de clase de funciones de editor

Este diagrama solo tendrá una ventana Edit. Esta clase va a ser la clase principal y contendrá todos los demás clases. Ahora vamos a ver el diagrama resultante en la figura 8 y posteriormente su descripción.

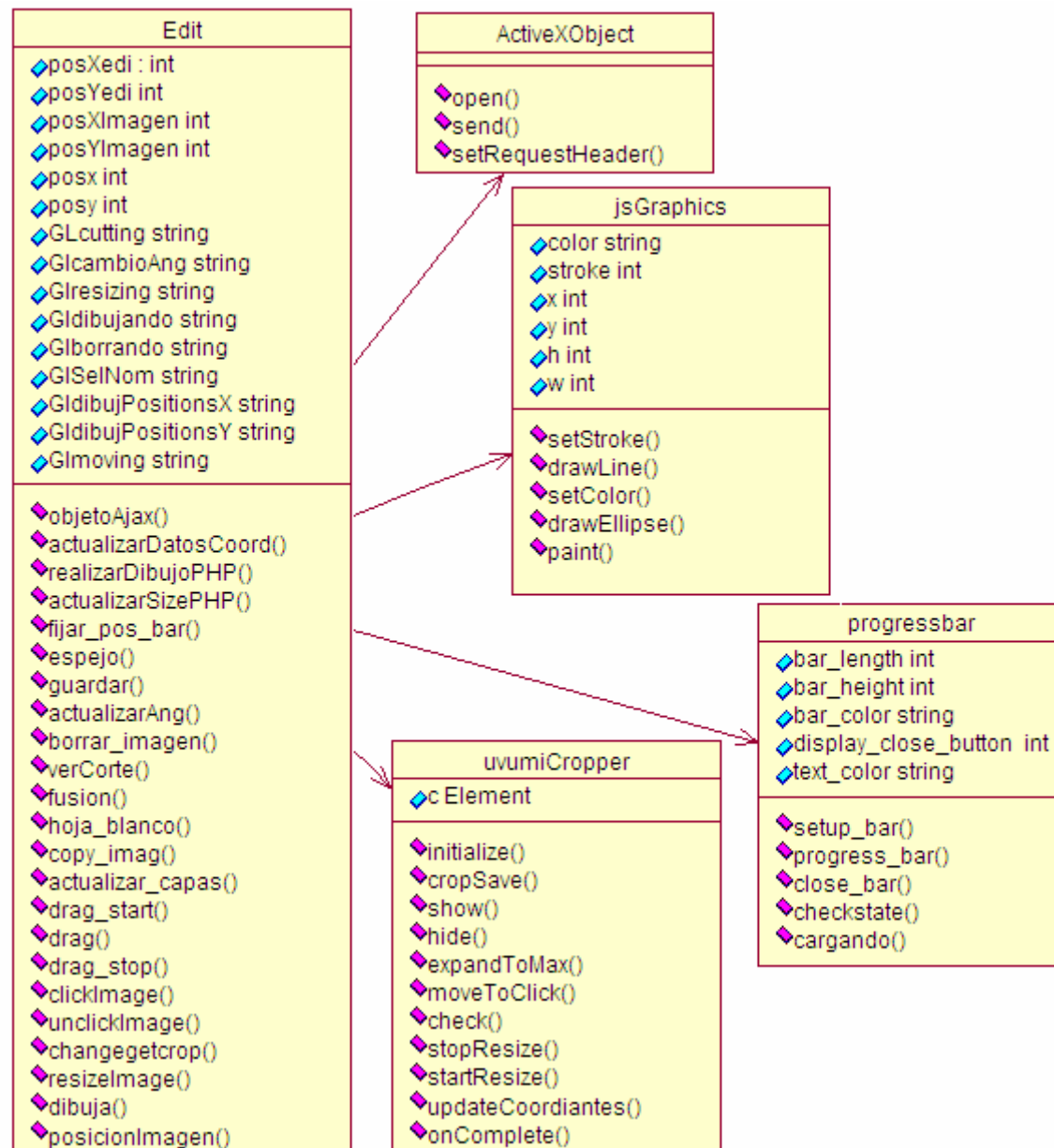


Fig.8

La clase ActiveXObject es para crear el objeto de AJAX. Nos va a servir mucho para pasar los datos de servidor a cliente sin recargar la página de cliente. Esto es necesario para pasar los órdenes de ventana de Edit a servidor y que el servidor ejecuta las órdenes que hay que realizar sobre una imagen y como resultado devolverá esa imagen a la página de cliente.

La clase de jsGraphics es para crear un objeto en la ventana de navegador de cliente para poder dibujar líneas o círculos.

La clase de progressBar, que contiene la ventana de Edit, es para enseñar al usuario cuanto tiempo le queda a esperar hasta que la ejecución de su orden será terminada.

Y última clase es uvumiCropper es para crear un objeto que sirve para recortar una área dentro de una imagen. Esta clase es reutilizada y no es necesario implementarla, solo habrá que realizar algún cambio.

Diagrama de clases de funciones especiales de administrador

Este diagrama está compuesto solamente por una clase. Es una ventana de eliminación de usuarios. Como esta ventana no tiene ninguna función en javascript, la clase tampoco va tener.

Separación por módulos

Vamos a utilizar la metodología funcional. Esta metodología trata de transformar las especificaciones del problema en una serie de módulos de forma que cada uno realice una función específica y al ejecutarse resuelven el problema.

Así pues vamos a dividir todas las funciones que deben existir en módulos:

Modulo de acceso: Este modulo contendrá funciones que están dirigidas a controlar el acceso de usuarios al sistema.

Modulo de gestión de imágenes: Sus funciones se encargaran de coleccionar las imágenes y luego mostrar todo lo que hay a un usuario. Este modulo incluye también las funciones de búsquedas de imágenes y categorías. Los usuarios registrados o no podrán hacer consultas y así pasar a modo de búsqueda después de cual la podrán acabar. También tiene funciones de fácil navegación por la página, usando los botones o un input para poder saltar a cualquier página directamente. Estando en alguna de las categorías de imágenes o estando en modo de búsqueda, las funciones de este modulo enseñaran el número total de páginas que hay en este caso. Además tiene funciones que sirven para simplificar la elección de una de las categorías. Los usuarios con funciones de este modulo tendrán posibilidad de borrar o editar las imágenes simplemente arrastradlas en una de las opciones. Sin embargo la funcionalidad de borrar solo permite a un usuario borrar solo los contenidos que fueron subidos por él.

Modulo de gestión de espacios de trabajos: Este módulos se encarga de coleccionar los espacios de trabajo y luego mostrar los a los usuarios. Además incluye los funciones de realización de búsquedas avanzadas, eso quiere decir que estas búsquedas pueden ser configuradas. También incluye funciones que permiten a los usuarios borrar o reemplazar su espacio de trabajo por uno de los elegidos, arrastrando las en una de las opciones. Este modulo contiene funciones parecidas o iguales al modulo anterior, tales como navegación por paginas. Sin embargo la funcionalidad de eliminación solo permite a un usuario borrar solo los contenidos que fueron guardados por él.

Modulo de edición de imágenes: Este modulo contendrá mucho mas funciones que los módulos anteriores. Estarán dirigidas a edición de imagen. Las funciones de edición se encargan de: la redimensión de imagen con la visualización de dimensiones actuales, el cambio de zoom del espacio de trabajo, el cambio de ángulo con visualización del ángulo que tendrá imagen, recortar imagen con visualización de área de recorte, hacer el efecto espejo de imagen, copiar una imagen, dibujar con la posibilidad de cambio de los colores y el tamaño, borrar con los diferentes tamaños, crear hojas en blanco, la fusión de todas las imágenes. Además tendrá las funciones de subida, eliminación y arrastramiento de las imágenes.

Modulo de administración de usuarios y contenidos web: Estas funciones son de administrador y le permiten eliminar cualquier usuario y su espacio de trabajo. Además el administrador podrá realizar la eliminación de cualquier imagen o espacio de trabajo dentro del sitio compartido.

El diagrama de módulos es la figura 8.

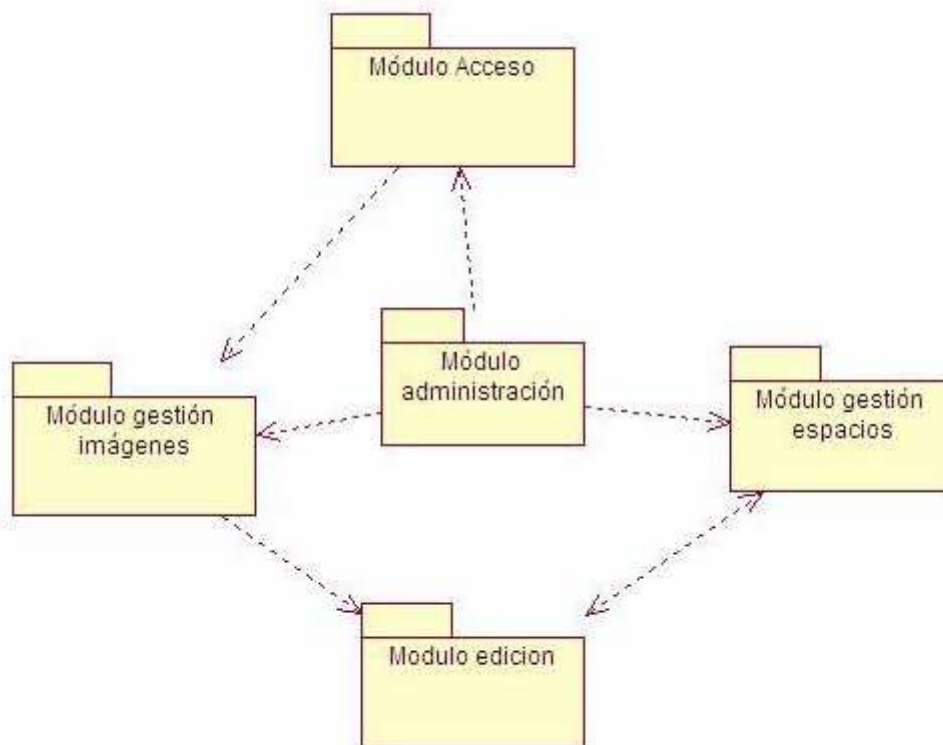


Fig.8

Así en el diagrama están todos los módulos que forman todo el sistema a diseñar.

Diagrama Entidad-Relación

Es evidente que tenemos que almacenar información en algún lugar. Como la base de datos es una solución más útil y sencilla que por medio de los ficheros, esta decido que usaremos la base de datos.

Según toda la información previa y los requisitos se diseño el siguiente diagrama de entidad – relación, ver la figura 9.

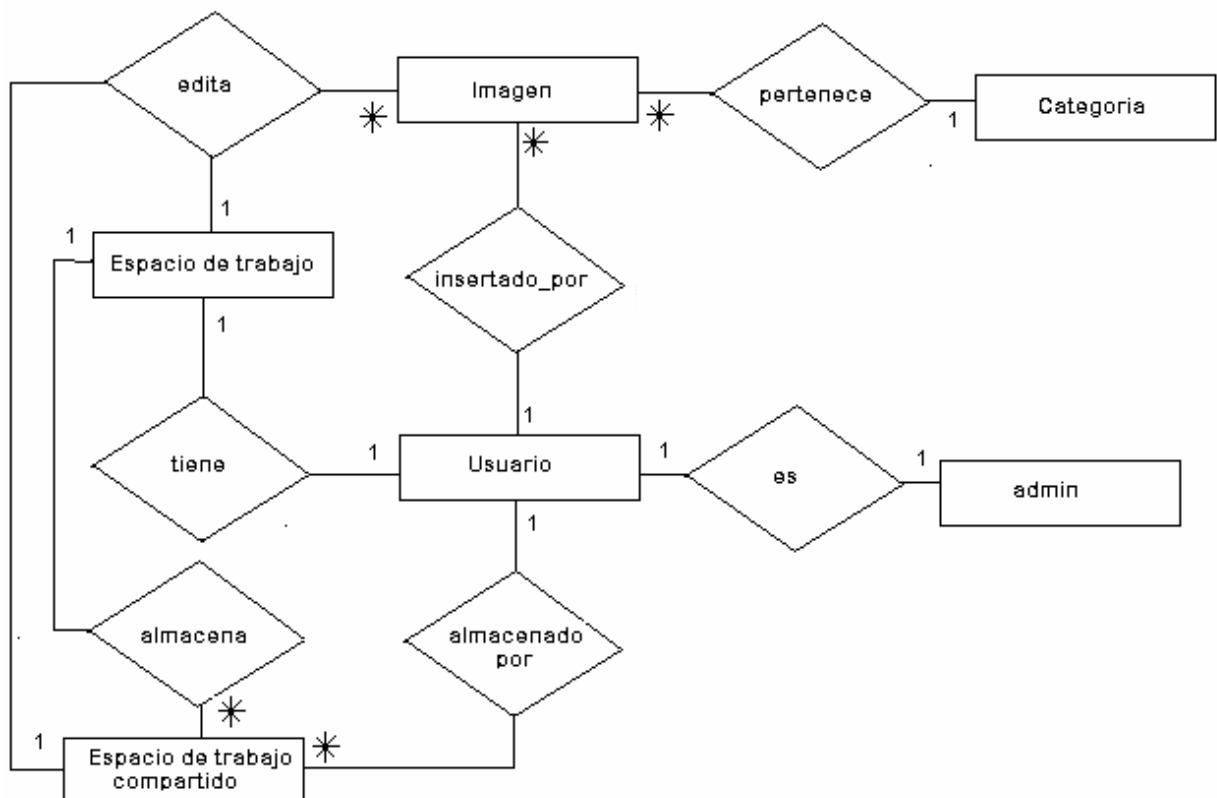


Fig.9

Las entidades son:

Imagen: es entidad agregada por el usuario que tendrá nombre de imagen único y su información asociada, podrá ser copiada a espacio de trabajo, por lo que el nombre de imagen se guarda.

Categoría: contiene las imágenes asociadas a ella. Y contendrá los nombres de categorías.

Admin: será uno de los usuarios de sistema.

Usuario: tendrá sus datos relacionados y podrá insertar las imágenes y almacenar los espacios de trabajo y tener los espacios de trabajo actuales.

Espacio de trabajo: es entidad de espacio donde un usuario tendrá sus imágenes guardadas y bajo desarrollo.

Espacio de trabajo compartido: Aquí estarán guardados los espacios de trabajo de los usuarios

Las relaciones existentes:

Insertado por: un usuario podrá insertar una cantidad cualquiera de imágenes

Pertenece: una imagen después de ser insertada pertenecerá a una de las categorías.

Es: un usuario puede ser un admin.

Almacenado por: un usuario podrá guardar sus espacios de trabajo actuales en el espacio de imágenes compartido.

Almacena: después de indicación por el usuario, el sistema almacena datos de espacio de trabajo actual de usuario.

Tiene: un usuario puede tener uno y solo uno espacio de trabajo.

Edita: el espacio de trabajo de un usuario o su espacio guardado en el sitio compartido podrá contener una o más imágenes.

3.1.2 DISEÑO E IMPLEMENTACIÓN

En este apartado se realizara el diseño del sistema y después se decidirán los pasos a seguir para implementar el código fuente.

Paso a tablas del diagrama ER

Vamos a describir las tablas que hemos conseguido a partir del diagrama entidad relación:

ADMINES

Atributo	Descripción	Presencia	Tipo	Refer.
<u>Nombre</u>	Supervisor de sistema	obligatoria	Varchar	
fecha_in	Fecha de inicio de administración	opcional	Date	

Esta tabla contendrá a todos los administradores del sistema. Un administrador también tiene que ser el usuario en la tabla de USER. También guardamos fecha para cada administrador, para saber desde cuando lleva administrando.

CODIGO_CATEGORIA

Atributo	Descripción	Presencia	Tipo	Refer.
<u>codigo</u>	Cada categoría tendrá un código asociado	obligatoria	Varchar	
nom_categoria	Nombres de categorías	obligatoria	Varchar	

Como estamos trabajando con la página web, las categorías están codificadas con los números para facilitar los saltos de una categoría a otra. Así pues en la tabla GALERIA_IMAG estarán los códigos de categorías y si queremos saber el nombre, tendremos que consultar esta tabla.

USER

Atributo	Descripción	Presencia	Tipo	Refer.
<u>nombre</u>	Cada usuario tendrá un nombre	obligatoria	Varchar	ADMINES(Nombr e)
pass	Contraseña para acceso	obligatoria	Varchar	

Esta tabla contiene a todos los usuarios registrados en nuestro sistema. También contiene a los administradores, serán los usuarios que tienen su nombre en la tabla de administradores.

GALERIA_IMAG

Atributo	Descripción	Presencia	Tipo	Refer.
<u>nombre</u>	Las imágenes serán nombradas nombre.jpg	obligatoria	Varchar	
orden	Este orden para cada categoría cuenta imágenes desde 1.	obligatoria	Integer	
categoría	Es el código de categoría	obligatoria	Integer	CODIGO_CATEGORIA(codigo)
orden_todo	Cuenta todas las imágenes desde 1.	obligatoria	Integer	
insertadoBY	Es el nombre de usuario que subió la imagen	opcional	Varchar	USER(nombre)

Esta tabla contiene todas las imágenes que están en gallería añadidos por los usuarios. Estas imágenes pueden ser cogidos y añadidos a tabla de EDIT_DATOS_IMAGEN para edición con lo cual el nombre de esa imagen puede ser un poco modificado.

Las imágenes se guardaran en **CARPETA** llamada **ALAMACEN**.

EDIT_DATOS_IMAGEN

Atributo	Descripción	Presencia	Tipo	Refer.
<u>nom_imag</u>	Es el nombre de imagen cogido desde el nombre original de gallería y modificado.	obligatoria	Varchar	
posX	Coordenada X ,posición de una imagen	obligatoria	Integer	
posY	Coordenada Y, posición de una imagen	obligatoria	Integer	
<u>prop_user</u>	El nombre de un usuario	obligatoria	Varchar	USER(nombre)
sizeX	Tamaño de la imagen	obligatoria	Integer	
sizeY	Tamaño de la imagen	obligatoria	Integer	

Ncapa	Posición de la imagen sobre o debajo de los demás imágenes	obligatoria	Integer	
angulo	Inclinación de una imagen	obligatoria	Integer	

Esta tabla tiene todos los espacios de trabajo que están en el proceso de desarrollo. Así un usuario podrá diseñar sus imágenes y luego salir, y luego volver a entrar y sus imágenes seguirán estando en su espacio de trabajo. Así pues en esta tabla cada usuario tiene un espacio de trabajo y en este espacio de trabajo ellos tienen varias imágenes.

Los espacios de trabajo se guardaran en la **carpeta ESPACIO DE USUARIOS**, en la carpeta llamada con el **NOMBRE DE USUARIO**

El espacio de trabajo está compuesto por: **imágenes, carpeta con imágenes originales y carpeta con imágenes redimensionadas.**

Además existirán carpetas como redim para la visualización y otra carpeta como small para el contenido de todas las imágenes redimensionadas con el pequeño tamaño para que una página web se cargue más rápido.

ESPACIOS_TRABAJO

Atributo	Descripción	Presencia	Tipo	Refer.
fecha	El día cuando el usuario había guardado su trabajo	obligatoria	Date	
nombre_espacio	El nombre con el que espacio de trabajo fue guardado por el usuario	obligatoria	Varchar	
<u>cod_dir_imag_ref</u>	El identificador de espacio	obligatoria	Integer	
nombre_user	El nombre de un usuario	obligatoria	Varchar	USER(nombre)

Esta tabla guarda información básica sobre cada espacio de trabajo almacenado por los usuarios. También guarda un código que tiene referencia a tabla ESPACIOS_TRABAJO_INFO, donde se guarda la información sobre las imágenes.

Estos espacios se guardan dentro de la **CARPETA DE SITIO COMPARTIDO**, en la carpeta que le corresponde según el número de cod_dir_imag_ref. Además, en esta carpeta se guardara la foto de espacio de trabajo, para poder ver el estado de las imágenes que contiene este espacio de trabajo.

ESPACIOS_TRABAJO_IMAGS_INFO

Atributo	Descripción	Presencia	Tipo	Refer.
<u>nom_imag</u>	Es el nombre de imagen ,cogido desde el nombre original de galería y modificado.	obligatoria	Varchar	
posX	Coordenada, la X posición de una imagen	obligatoria	Integer	
posY	Coordenada Y,la posición de una imagen	obligatoria	Integer	
<u>codigo_ref_espacio</u>	El código de espacio de trabajo	obligatoria	Integer	ESPACIOS_TRABAJO(co d_dir_imag_ref)
sizeX	Tamaño de la imagen	obligatoria	Integer	
sizeY	Tamaño de la imagen	obligatoria	Integer	
Ncapa	Posición de la imagen sobre o debajo de los demás imágenes	obligatoria	Integer	
angulo	Inclinación de imagen	obligatoria	Integer	

Esta tabla guarda la información sobre todas las imágenes de cada uno de los espacios de trabajo almacenados en el sitio compartido. Así pues, cuando un usuario guarda su espacio de trabajo actual, todos sus datos que están en la tabla ESPACIOS_TRABAJO se copian a esa tabla.

CONSTANTES

Atributo	Descripción	Presencia	Tipo	Refer.
sec_orden_espacios	Cada vez que se guarda un espacio de trabajo le tenemos que asignar un nuevo código.	Obligatoria	Integer	

Cada vez cuando se almacena un nuevo espacio de trabajo en el sitio compartido, le tenemos que asignar un código nuevo y este tiene que ser mayor de los otros ya escogidos. Así cuando se guarda un espacio de trabajo, se consulta el valor de `sec_orden_espacio` e se inserta en el `cod_dir_imag_ref` de la tabla ESPACIOS de TRABAJO. Teníamos que hacer esta tabla porque con tan solo hacer el max de todos los espacio de trabajo que hay, no funcionaria bien, porque cuando se elimina un espacio de trabajo, por ejemplo el de `cod_dir_imag_ref` de valor máximo y justo después se guarda un nuevo espacio de trabajo, entonces este tendrá el mismo valor que se elimino antes, y el navegador a la hora de enseñar el nuevo espacio de trabajo, enseñará la imagen de espacio de trabajo eliminado, porque la cache del navegador no se reinicia.

Elección de la tecnología a usar

En ese apartado se indicara que herramientas y lenguajes de programación fueron usados para implementar el sistema operativo. Luego se explicaran los motivos por lo que estas herramientas fueron escogidas.

El ordenador de cliente debe tener instalado un navegador Firefox (v3.0, v3.5, v3.6) o Internet Explorer (v7, v8) con las habilitaciones de:

AJAX

Javascript

Esta aplicación se realizo en un ordenador (Servidor) equipado con:

Ubuntu 9.04

Apache

MySQL

Intérprete de PHP (con ImageMagick y GD library) y HTML

El sistema operativo es Ubuntu 9.04, pero podría ser cualquier otra distribución de Linux. Esta plataforma es elegida por tener mucha más experiencia de trabajo con ella que con Windows.

Para trabajar con la base de datos hemos elegido MySQL. Porque es muy fácil de usar, además hay una aplicación llamada phpmyAdmin que facilita el trabajo con MySQL y es muy útil para gestionar la base de datos. MySQL es un sistema de gestión de la base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL es muy utilizado en las aplicaciones web, su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

Para desarrollar las aplicaciones de este tipo solo hay dos opciones para elegir es hacer lo en javascript o en flash. Hemos elegido javascript por tener más experiencia con este lenguaje de programación.

PHP es un lenguaje para la creación de páginas web dinámicas y puede ser incrustado dentro de código HTML. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Nuestra aplicación va a ser compuesta por las páginas web dinámicas. Además hemos integrado ImageMagick y GD library en PHP, instalando en PHP las extensiones correspondientes para poder usar sus funciones de librería.

También hemos elegido como el servidor Apache, porque Apache presenta entre otras características altamente configurables, las bases de datos de autenticación y negociado de contenido y Apache, es el servidor HTTP más usado.

AJAX, acrónimo de Asynchronous JavaScript And XML , es una técnica de desarrollo web para crear aplicaciones interactivas . Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

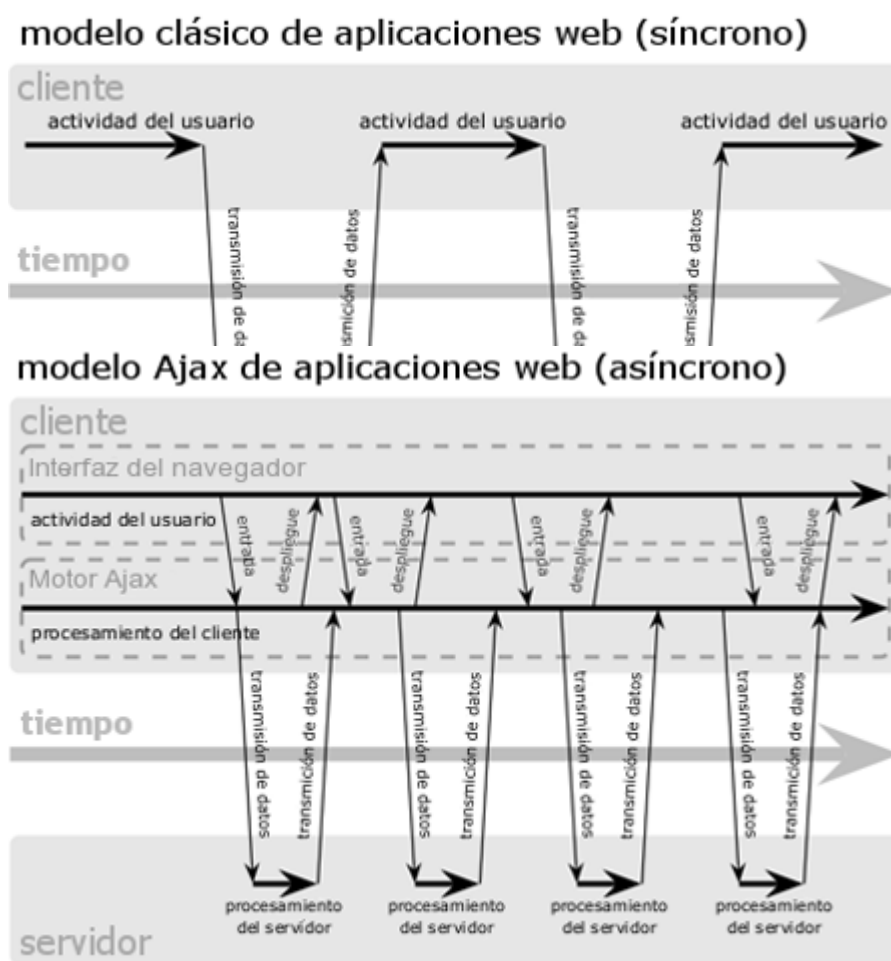


Fig.10

Javascript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. Así pues, nos viene bien esta tecnología porque necesitamos que solo se recargase una imagen después de haber realizado una operación sobre ella y no la recarga de toda la página web, ver la figura 10.

Módulos

En el análisis hemos determinado los módulos y ya sabemos que hace cada uno de ellos. Ahora queda por saber cómo funcionan. Para poder verlo hay que describir cada una de las funciones de cada módulo. Las funciones menos significativas pueden ser omitidas.

A partir de aquí se explicaran las funciones que tiene cada módulo:

MÓDULO ACCESO

Su nombre es **inscrip.php**. Tiene dos archivos php, uno es principal **inscrip.php** y el otro es **InscrEnBD.php** y es llamado con el formulario (FORM).

Aquí se describirán todas las funciones menos los de bola giratoria que es la parte de código reutilizado. Esta bola simplemente sirve para selección de una categoría, también tenemos otra alternativa de selección, es a través de menú.

Llamada desde	inscrip.php
función	InscrEnBD.php
Entrada	Nom, pas1, pas2 String
Salida	Devuelva en caso de una mala introducción de datos un aviso de fallo.
Descripción	Realiza conexión a la base de datos y comprueba si los datos enviados desde inscrip.php a través de GET son correctos y si todo está bien entonces inserta el nuevo usuario en BD.
Errores	Excepción de Base de datos

Llamada desde	Todos los módulos menos el de edición
función	Login.php , Login2.php
Entrada	Nom, pas String
Salida	Devuelva en caso de mala introducción de datos un aviso de fallo.
Descripción	Accede a la base de datos y comprueba si el usuario existe y el pass es correcto. Los dos scripts son parecidos, pero uno después de realizar una operación salta a la página de galería y el otro a la página de espacios.
Errores	Excepción de Base de datos

Llamada desde	Todos los módulos menos el de edición
función	destuctsession.php, destuctsession2.php
Descripción	Equivala a hacer LOGOUT. Los dos scripts son parecidos, pero uno después de realizar una operación salta a la página de galería y el otro a la página de espacios.

MÓDULO ADMINISTRACIÓN

Su nombre es **borrarUser.php**. Tiene dos archivos php, uno es principal **borrarUser.php** y el otro es **borrarUSERdeBD.php** y es llamado con el formulario (FORM).

Llamada desde	borrarUser.php
función	borrarUSERdeBD.php
Entrada	Nom, pas1, pas2 String
Salida	Devuelva en caso de mala introducción de datos un aviso de fallo.
Descripción	Realiza conexión a la base de datos y comprueba si los datos enviados desde borrarUser.php a través de GET son correctos y si todo está bien entonces borra el usuario.
Errores	Excepción de Base de datos

También en este modulo se incluyen las funciones de **borrar imágenes y espacios** pero se ha decidido que estarán en modulo de Gestión de espacios y imágenes.

MÓDULO GESTIÓN DE IMÁGENES

Su nombre es **entrada.php**. Este modulo está compuesto por funciones en javascript y php.

Llamada desde	Onmousedown Evento de las Imágenes
función	drag_start
Entrada	ItemToMove Object, e Event
Descripción	Realiza inicialización de las coordenadas iniciales, de capas y de otros valores de las imágenes.

Llamada desde	Onmousemove Evento del documento
función	drag
Entrada	e Event
Descripción	Determina las coordenadas de objeto cuyo evento fue producido y cambia las coordenadas de este objeto por los nuevos valores. El objeto en ese caso es una imagen.

Llamada desde	Onmouseup Evento del documento
función	drag_stop
Entrada	e Event
Descripción	Parar de mover una imagen.

Llamada desde	Evento onmouseover de las imágenes
función	clickImage
Entrada	idImage String
Descripción	Por el nombre de id de imagen cogemos el objeto de la imagen y le cambiamos los bordes y además guardamos en variable global el nombre de imagen seleccionada.

Llamada desde	Evento onmouseout de las imágenes
función	unclickImage
Entrada	idImage String
Descripción	Deselecciona una imagen.

Llamada desde	Onmouseup de imagen de reciclaje y edición
función	posicionImagen
Entrada	Imagen Object
Salida	Actualiza las posiciones de imagen de reciclaje o de edición.
Descripción	Determina la posición de una imagen

Llamada desde	función eliminarOno
función	getMouse
Entrada	e Event
Salida	Actualiza las variables globales de la posición de ratón.
Descripción	Determina la posición de una imagen cuando se produce el evento.

Llamada desde	Evento onclick del documento
función	eliminarOno
Entrada	elEvento Event
Descripción	Comprueba si la imagen seleccionada esta sobre la imagen de reciclaje o de edición y si todo va bien llama a borrarIm.php o a subimagtoedit.php respectivamente pasándoles el nombre de esa imagen.

Llamada desde	Evento ondbclick de la imagen de reciclaje
función	editanonewing
Descripción	Abre la ventana de edición. Si la carpeta de espacio de usuario no existe la crea.

función	abre
Salida	Ventana Objeto
Descripción	Abre una página de EDICION

Llamada desde	Evento ondbclick de las imágenes.
función	VerTamReal
Descripción	Abre la ventana con esa imagen en grande.

Llamada desde	Evento onchange de lista de categorías
función	MM_jumpMenu
Entrada	targ String; selObj Object;restore Int
Salida	Lista construida con las categorías.
Descripción	Construye una lista con categorías en forma de menú.

Llamada desde	Formulario de búsqueda
función	busqueda.php
Entrada	Abuscar String
Descripción	Inicia valores de búsqueda en sesión y salta a la página de entrada.php. Estamos en modo de búsqueda. Solo salen las imágenes con los nombres buscados.

Llamada desde	Formulario de entrada.php
función	subimag.php
Entrada	CATEGORIA, USER String
Descripción	Guarda imagen en la base de datos y en carpeta de almacén en la categoría que le corresponde.

Llamada desde	Formulario de entrada.php
función	subimagtoedit.php
Entrada	c_imToedit, USER, newimagen, c_path_edit String
Descripción	Guarda la imagen en el directorio de Espacios de usuarios en la carpeta de usuario con el nombre USER.

Llamada desde	entrada.php
función	redim.php
Entrada	Img String
Salida	Imagen redimensionada
Descripción	Sirve para redimensionar la imagen y guardarlo en carpetas redim y small, para eficiencia de la carga de página.

Llamada desde	entrada.php
función	acabarBusqueda.php
Descripción	Acaba el modo de búsqueda y vuelve a saltar a la página de entrada.

función	set_cookie
Entrada	Name, value String
Salida	Cookie creado y iniciado con una valor
Descripción	Sirve para crear e iniciar un cookie.

función	leer_cookie
Entrada	Name String
Salida	Valor de cookie
Descripción	Sirve para leer un cookie.

Llamada desde	entrada.php (eliminarOno)
función	borrarIm.php
Entrada	c_imTodrop String
Descripción	Elimina una imagen de galería con el nombre c_imTodrop.

Llamada desde	entrada.php (eliminarOno)
función	borrarIm.php
Entrada	c_imTodrop String
Descripción	Elimina una imagen de galería con el nombre c_imTodrop.

MÓDULO GESTIÓN DE ESPACIOS

Este modulo es **espacios_trabajo.php**. Este modulo tiene funciones iguales a los de módulos de GESTION DE IMAGENES. Las funciones iguales y descritas ya, son:

drag_start
drag_stop
drag
set_cookie
getMouse
abre
posicionImagen=posicionImagenDespacio
clickImage=clickImageDeEspacio
unclickImage=unclickImageDeEspacio
posicionImagen=posicionImagenDEesp

Llamada desde	Evento onclick de documento
función	eliminarOno
Entrada	ElEvento Event
Descripción	Comprueba sobre qué imagen esta la imagen de espacio de trabajo, sobre edición o eliminación. Si eliminación entonces llama a del_espacio_trab.php sino a gestficheditREEMPAZO.php o guardarEspacio.php .

Llamada desde	espacios_trabajo.php (eliminarOno)
función	del_espacio_trab.php
Entrada	c_espTodrop , USER String
Descripción	Si el espacio es de nombreUSER usuario entonces elimina el espacio de trabajo con el nombre c_espTodrop.

Llamada desde	espacios_trabajo.php (eliminarOno)
función	gestficheditREEMPAZO.php
Entrada	NameSAVEespacio, guardarEspacio, codref, user String
Descripción	Si guardarEspacio=yes entonces guardamos el espacio de usuario en el sitio compartido con el nombre NameSAVEespacio. Luego reemplazamos su espacio de trabajo por el espacio que tiene en la base de datos donde cod_ref= codref.

Llamada desde	Ondbclick de las imágenes.
función	editespacio
Descripción	Hace que se abra la ventana de edición llamando a inicioEditFromEsp.php .

Llamada desde	espacios_trabajo.php (editespacio)
función	inicioEditFromEsp.php
Descripción	Crea las carpetas para el espacio de trabajo de usuario actual, si no existen. Y abre la ventana de edición.

Llamada desde	espacios_trabajo.php (formulario)
función	busquedaEspacios.php
Entrada	Orderdate Date; busquedaNombreESP, busquedaNombreUSR String
Descripción	Inicia los valores y vuelve a saltar a espacios_trabajo.php enseñando solo los espacios que cumplan los requisitos de búsqueda.

Llamada desde	espacios_trabajo.php
función	acabarBusquedaEspac.php
Descripción	Acaba el modo de búsqueda y vuelve a saltar a la página de espacios.

MODULO EDICIÓN

Este modulo es **edicion.php**.

Este modulo contiene funciones iguales a las de módulos de GESTIÓN DE IMÁGENES. Las funciones iguales y descritas ya son:

Las funciones se dividen según el tipo de trabajo que realizan:

Funciones COMUNES:

posicionImagen

getMouse

set_cookie

Funciones para SELECCIONAR imagen:

unclickImage

clickImage

Funciones para MOVER imagen:

drag_stop

drag_start

drag

Funciones para TRABAJAR CON AJAX:

función	objetoAjax
Salida	Xmlhttp XMLHttpRequest
Descripción	Devuelve un objeto XMLHttpRequest o ActiveXObject dependiendo de sistema operativo.

Funciones para MOVER imagen:

Llamada desde	Evento onmouseup de las Imágenes
función	actualizarDatosCoord
Entrada	imagen Object
Descripción	Llama por AJAX a actualizaPos.php que guarda en base de datos la nueva posición de imagen.

Llamada desde	actualizarDatosCoord
función	actualizaPos.php
Entrada	NomImag String; posx,posy Int
Descripción	Recibe los datos a través de ajax por POST y actualiza nueva posición de imagen en la BD.

Llamada desde	drag_start
función	actualizar_capas
Entrada	Imagen seleccionada y zMax capa máxima globales
Descripción	Manda por Ajax el nombre de imagen seleccionada y el capa máxima a actualizar_capas.php .

Llamada desde	actualizar_capas
función	actualizar_capas.php
Entrada	USER,nomImag String;newcapa Int
Descripción	Recibe por ajax el nombre de imagen y de quien es imagen por SESSION y actualiza esta imagen cambiando la capa de esa imagen por newcapa.

Llamada desde	Onmouseover Evento de las Imagenes
función	indicar
Entrada	IdImage String
Descripción	Selecciona la imagen con color verde cuando el ratón pasa sobre la imagen deseleccionada.

Llamada desde	Onmouseout Evento de las Imágenes
función	stopindicar
Entrada	IdImage String
Descripción	Hace lo contrario a lo que hace función indicar.

Funciones para **DIBUJAR** sobre imagen:

Llamada desde	Event onmousemove de imagen si estamos dibujando
función	dibuja
Entrada	Evt Event; tiponaveg Int
Uso de librerías o clases	JsGraphics clase;
Descripción	Recibe Evt evento de imagen para averiguar las coordenadas de ratón, guardando lo en los variables globales GIIdibujPositionsX,Y y tiponaveg para el ajuste de dibujo. Y dibuja en el navegador el dibujo, creando el objeto jg_dibuja = new jsGraphics(cnv_d); , y usando las funciones de la clase jsGraphics: drawLine, setColor, setColor, paint.

Llamada desde	Event onmouseup de imagen si estamos dibujando
función	realziarDibujoPHP
Uso de librerías o clases	progress.js
Descripción	Averigua el color col y el grosor seleccionado, pasando los datos por AJAX a actualizaDibujo.php . El resto de datos pasados son globales. Después vacía GDibujPositionsX,Y , son datos de coordenadas de líneas a dibujar. Activa una barra de espera para saber cuándo imagen, enviada por actualizaDibujo.php , finaliza a descargarse.

Llamada desde	realziarDibujoPHP
función	actualizaDibujo.php
Entrada	NomImag, selctdName,colorLine String; PosImagX,PosImagY,size, anch, al, vectPosX, vectPosY Int
Uso de librerías o clases	GD library funciones usadas: imagecreatefromjpeg, imagejpeg, imagecreatefromjpeg, imagesetthickness, imageline, imagedestroy
Descripción	Recibe los datos a través de ajax y actualiza el dibujo sobre la imagen. Toma las coordenadas de las líneas a dibujar y las dibuja usando funciones de librería de GD.

Funciones para **VOLVER A MOVER** imagen:

Llamada desde	Event onmousedown sobre el botón
función	VolverAmover
Descripción	Comprueba si la operación actual es rotar, dibujar o borrar y hace que esta operación se acaba. En caso de que si es operación de dibujar o borrar se recarga la imagen.

Llamada desde	VolverAmover
función	cargarImagenSimple
Uso de librerías o clases	progress.js
Descripción	Crea el objeto para Ajax y luego llama a cargarImagenSimple.php , que recarga la imagen modificada con operación de dibujar. También activa la barra de espera para saber cuándo termina la recarga de esa imagen enviada por cargarImagenSimple.php .

Llamada desde	cargarImagenSimple
función	cargarImagenSimple.php
Entrada	NomImag, selctdName String
Salida	Imagen que se pasa por AJAX a Edicion.php
Descripción	Recibe los datos a través de ajax por POST. Los datos son el nombre de imagen y el identificador de la imagen seleccionada sobre cual estamos dibujando, borrando o rotando. Simplemente el Ajax hace que se recarga esa imagen en el navegador.

Funciones para **CAMBIO DE ÁNGULO** de imagen:

Llamada desde	Event onmousedown sobre el botón
función	cambioAng
Descripción	Llama a cargaImagForAngulo luego a retardo_para_carga .

Llamada desde	cambioAng
función	CargaImagForAngulo
Uso de librerías o clases	progress.js
Descripción	Crea el objeto para Ajax y luego llama a cargaImagPreCambioAngul.php , que recarga la imagen original. También activa la barra de espera para saber cuándo termina la recarga de esa imagen enviada por cargaImagPreCambioAngul.php .

Llamada desde	CargaImagForAngulo
función	cargaImagPreCambioAngul.php
Entrada	NomImag, selctdName String
Salida	Imagen original que se pasa por AJAX a Edicion.php
Descripción	Hace que la imagen que se va a rotar se recargue en navegador. Es posible que hemos rotado por la primera vez la imagen y queremos rotar la misma imagen otra vez. Así que necesitamos la imagen original, es la imagen que estaba antes de haber hecho la primera rotación.

Llamada desde	cambioAng
función	retardo_para_carga
Uso de librerías o clases	progress.js
Descripción	Esta función espera a que la imagen esté completamente cargada, llamando a funciones de progress.js , y después llama a cargaAcabada .

Llamada desde	retardo_para_carga
función	cargaAcabada
Descripción	Esta función prepara todos los valores que necesita la función dibujar_recta_angul . Esos valores dependen de posición, tamaño de imagen y posición relativa de ratón. Y luego llamara a dibujar_recta_angul .

Llamada desde	cargaAcabada,checkArea
función	dibujar_recta_angul
Entrada	Ax,Ay,Bx,By,MouseX,MouseY,PosImagX,PosImagY,TamImagX,TamImagY Int
Uso de librerías o clases	JsGraphics clase;
Descripción	Usa los valores preparados antes en función cargaAcabada o checkArea , para dibujar el ángulo grafico sobre la imagen para poder seleccionar un ángulo necesario a cambiar, usando la clase jsGraphics .

Llamada desde	Evento onmousemove de las Imágenes
función	checkArea,checkArea2(Explorer)
Entrada	ImagObj Object, name String,evt Event
Descripción	Si estamos en proceso de cambio de ángulo, entonces esta función prepara los valores necesarios que dependen de posición del ratón y la imagen y llama a dibujar_recta_angul para refrescar la imagen grafica que enseña el dibujo que permite ver la inclinación. Si no estamos en ningún proceso de edición como dibujando o cambiando ángulo, entonces comprueba si el ratón está en la esquina derecha de imagen, y si es así pasamos a proceso de redimensionamiento.

Llamada desde	Evento onmousedown de las Imágenes(si estamos en proceso de cambio de ángulo)
función	actualizarAng
Uso de librerías o clases	progress.js
Descripción	Toma el ángulo que estaba en proceso de selección hasta ahora y lo pasamos por AJAX a cambioAngulo.php . También activa la barra de espera para saber cuándo termina la recarga de la imagen con nuevo ángulo, enviada por cambioAngulo.php .

Llamada desde	actualizarAng
función	cambioAngulo.php
Entrada	angulo Int; nomImag, selctdName String
Salida	Imagen con el nuevo ángulo
Uso de librerías o clases	Librería de ImageMagick → imagick clase métodos usadas: new Imagick,rotateImage,writeImage
Descripción	Recibe los datos :el nombre de imagen, id y el ángulo y cambia el ángulo de esta imagen, usando las funciones de librería de ImageMagick .

Llamada desde	CheckArea2,checkArea,dibujar_recta_angul
función	checkIntersection
Entrada	x1,y1,x2,y2,x3,y3,x4,y4 Int
Descripción	Se usa para saber el punto de intersección de dos líneas. Guarda el resultado en variables globales. Se usa para hacer el dibujo para poder elegir el ángulo.

Funciones para **REDIMENSIONAR** imagen:

Llamada desde	Evento onmousedown de las Imágenes
función	active,active2(Explorer)
Descripción	Comprueba en que parte de imagen hemos pulsado el botón, si es en la esquina derecha de imagen entonces llama a changetoresize .

Llamada desde	active,active2
función	changetoresize
Descripción	Pone el proceso actual a redimensión cambiando la variable global.

Llamada desde	Evento onmousemove de las Imágenes (si estamos en proceso de redimensión)
función	resizeImage,resizeImage2(para Explorer)
Entrada	Evt Evento;pos_raton_newX,pos_raton_newY Int,GLnameresizing int
Descripción	Inicia la visualización de la información con setInterval de tamaño de imagen llamando a función actualiPosInfoSize() , y cambia el tamaño de imagen con nombre GLnameresizing por otro tamaño dependiendo donde se encuentra el ratón. Solo se cambia el tamaño de imagen en el navegador.

Llamada desde	ResizeImage,resizeImage2(para Explorer)
función	actualiPosInfoSize
Entrada	Posx,posy Int
Descripción	Enseña una información de tipo 100x100 de tamaño de imagen en la posición donde se encuentra el ratón posx y posy.

Llamada desde	Evento onmouseup de las Imágenes (si estamos en el proceso de redimensión)
función	actualizarSizeImag
Descripción	Toma las dimensiones nuevas de imagen seleccionada y pasa los datos por AJAX a actualizaSize.php . Activa la barra de estado de descarga para saber cuándo termina la recarga de imagen enviada por actualizaSize.php .

Llamada desde	actualizarSizeImag
función	actualizaSize.php
Entrada	USER,nomImag,selctdName String; anch,al Int
Salida	Imagen redimensionada
Uso de librerías o clases	GD library funciones usadas: getimagesize,imagejpeg,imagepng,imagegif, imagecopyresampled
Descripción	Recibe los datos a través de ajax por POST y actualiza anchura y altura de imagen nomImag de usuario USER que le llega por SESSION, usando la GD library .

Funciones para **BORRAR** imagen:

Llamada desde	eliminarOno
función	borrar_imagen
Entrada	NimImag String
Descripción	Llama a borrarImedit.php pasando le el nombre de imagen.

Llamada desde	borrar_imagen
función	borrarImedit.php
Entrada	c_imTodrop,USER String
Descripción	Recibe por ajax el nombre de imagen a borrar y por SESSION de quien es imagen y lo borra en la BD y en la carpeta de usuario

Funciones para **CREAR IMAGEN EN BLANCO**:

Llamada desde	Evento Onmousedown desde el botón
función	hoja_blanco
Entrada	Posxmouse, posymouse int
Uso de librerías o clases	progress.js
Descripción	Manda por Ajax la posición del ratón y la capa máxima a blancaHoja.php . Activa la barra de estado de descarga para saber cuándo termina la carga de imagen en blanco enviada por actualizaSize.php .

Llamada desde	hoja_blanco
función	blancaHoja.php
Entrada	Posxmouse,posymouse,newcapa Int;USER String
Salida	Imagen en blanco que se pasa por AJAX a Edicion.php
Descripción	Copia una imagen en blanco, que está en el carpeta usaImag , en el espacio de trabajo de usuario, y la asigna a este usuario de SESSION con posiciones posxmouse,posymouse y con capa newcapa en la BD.

Funciones para **COPIAR** imagen:

Llamada desde	Evento Onmousedown desde el botón
función	copy_imag
Entrada	posxmouse,posymouse int
Uso de librerías o clases	progress.js
Descripción	Manda por Ajax la posición del ratón, la capa máxima y el nombre de imagen seleccionada a copyImag.php . Y activa la barra de estado de descarga para saber cuándo termina la carga de imagen enviada por copy_imag.php .

Llamada desde	copy_imag
función	copy_imag.php
Entrada	Posxmouse,posymouse,newcapa Int; USER,nomselec String
Salida	Imagen copiada que se pasa por AJAX a Edicion.php
Descripción	Hace una copia de imagen con nombre nomselec y la asigna en la BD al usuario de SESSION con los posiciones Posxmouse,posymouse y con capa newcapa.

Funciones para **RECORTAR** imagen:

Llamada desde	Evento Onmousedown desde el botón
función	chanegetocrop
Salida	Objeto de uvumiCropper
Uso de librerías o clases	uvumiCropper clase;
Descripción	Creamos el objeto de clase uvumiCropper . Este clase tiene métodos para poder seleccionar la área de imagen y luego recortar la.

Llamada desde	uvumiCropper(OnCropSuccess Event)
función	verCorte
Descripción	Cuando ya hemos usado la clase uvumiCropper , para recortar la imagen y guardar la en el servidor, ahora tendremos que guardar la en base de datos y pasar la por ajax a la ventana de edición. Manda por Ajax la posición de ratón, la capa máxima a cargar_imag_cortada.php .

Llamada desde	verCorte
función	cargar_imag_cortada.php
Entrada	posxmouse, posymouse, newcapa Int
Salida	Imagen recortada que se pasa por AJAX a Edicion.php
Descripción	Guardamos los datos de imagen recortada en la base de datos y la pasamos por ajax a la ventana de edición.

Funciones para **FUSIONAR** imagen:

Llamada desde	Evento Onmousedown desde el botón
función	fusion
Entrada	posxmouse,posymouse int
Uso de librerías o clases	progress.js
Descripción	Manda por Ajax la posición del ratón, la capa máxima a marge.php . Activa la barra de estado de descarga para saber cuándo termina la carga de imagen enviada por marge.php .

Llamada desde	fusion
función	marge.php
Entrada	Posxmouse,posymouse,newcapa Int; USER String
Salida	Imagen fusionada que se pasa por AJAX a Edicion.php
Uso de librerías o clases	GD library funciones usadas: imagecreatetruecolor,imagecolorallocate,getimagesize, imagecreatefromgif(jpeg,png),imagecopymerge,

	imagedestroy,imagejpeg(png,gif)
Descripción	Hace una fusión de todas las imágenes de usuario USER y las asigna en BD a usuario de SESSION con posiciones. Posxmouse,posymouse y con capa newcapa.

Funciones para **VOLTEAR HORIZONTALMENTE** imagen:

Llamada desde	Evento Onmousedown desde el botón
función	espejo
Entrada	posxmouse,posymouse int
Descripción	Manda por Ajax la posición del ratón, la capa máxima y el nombre de imagen seleccionada a espejoHorizontal.php . Activa la barra de estado de descarga para saber cuándo termina la carga de imagen enviada por espejoHorizontal.php .

Llamada desde	espejo
función	espejoHorizontal.php
Entrada	USER,nomselec,nomImag,selctdName String
Salida	Imagen volteada que se pasa por AJAX a Edicion.php
Uso de librerías o clases	Librería de ImageMagick → imagemagick clase métodos usados: new Imagick ,writeImage,flopImage
Descripción	Voltea la imagen con el nombre selctdName.

Funciones para **INICIAR VALORES A LA HORA DE CARGA DE LA PÁGINA:**

Llamada desde	Evento onload de documento
función	iniciarvalores
Descripción	Ejecuta por Ajax el script inicio_valors.php .

Llamada desde	iniciarvalores
función	inicio_valors.php .
Salida	Devuelve maxcapa por cookie
Descripción	Hace una consulta en base de datos para saber la capa máxima y la devuelve a Edicion.php para iniciar zMax .

Funciones para **GUARDAR** espacio de trabajo:

Llamada desde	Evento Onmousedown desde el botón
función	guardar
Descripción	Pregunta a usuario el nombre de espacio a guardar llamando a función show_prompt(función simple con msnbox consulta) . Crea el objeto para Ajax y luego llama a guardar.php que guarda todo el espacio de trabajo.

Llamada desde	guardar
función	guardar.php
Entrada	nomEsp String
Uso de librerías o clases	GD library funciones usadas: imagecreatetruecolor, imagecolorallocate, getimagesize, imagecreatefromgif(jpeg,png), imagecopymerge, imagedestroy, imagejpeg(png,gif)
Descripción	Recibe por ajax el nombre de espacio. Hace fusión de todas las imágenes de este espacio de trabajo y lo guarda en el sitio compartido.

Funciones para **CERRAR EDITOR CUANDO SE CIERRA LA SESIÓN:**

Llamada desde	Event setInterval('comprobSesion()',1666)
función	comprobSesion
Descripción	Comprueba si userclosed esta iniciado, si es así se acaba edición y si no se llama a comprobSESSIONajax .

Llamada desde	comprobSESSIONajax
función	comprobacionSESSION.php
Descripción	Comprueba si la sesión de usuario sigue iniciada y que el usuario no se ha cambiado de sesión (iniciando con otra cuenta) .Si no es así entonces iniciamos por cookie la variable userclosed .

Funciones para **SABER EL TIPO DE NAVEGADOR** imagen:

Llamada desde	Edicion.php
función	ObtenerNavegador
Entrada	user_agent String
Salida	navegador String
Descripción	Para saber qué tipo de navegador tiene el cliente.

Esta clase se usa por las funciones de **DIBUJAR, CAMBIO DE ÁNGULO:**

CLASE **jsGraphics** (*lib/wz_jsgraphics.js*)

DESCRIPCION:

Esta clase se usa para poder dibujar en el navegador.

USO:

Crear objeto con: **new jsGraphics(id de div)**

método	setColor
Entrada	nextcol String
Descripción	Fija el nuevo color de pincel.

método	setStroke
Entrada	Nextsize Int
Descripción	Fija el grosor de pincel.

método	drawLine
Entrada	x1,y1,x2,y2 Int
Descripción	Dibuja la línea con coordenadas x1, y1, x2, y2.

método	drawEllipse
Entrada	CentrX, CentrY, width, height Int
Descripción	Dibuja el elipse en posición CentrX, CentrY con dimensiones width, height .

método	paint
Descripción	Todas las operaciones hasta ahora (como drawLine) dibujaban en el “buffer” ahora para sacarlo en pantalla hay que llamar a ese método.

método	clear
Descripción	Después de haber llamado a paint es necesario vaciar el buffer clear antes de llenarlo otra vez.

Esta clase se usa por las funciones de **RECORTAR**:

CLASE `uvumiCropper(lib/UvumiCrop-compressed.js)`

DESCRIPCION:

Esta clase se usa para poder recortar la imagen.

USO:

Clear objeto con: `new uvumiCropper(idImageSelected,{ coordinates:true,
preview:true,
cancelButton:true,
saveButton:true});`

método	destroy
Descripción	Destruye a sí mismo.

método	hide
Descripción	Esconde el objeto sin destruirlo.

método	show
Descripción	Después de haber escondido ,volvemos a enseñar.

método	changeImage
Entrada	Url String
Descripción	Actualiza la imagen con la nueva url.

método	cropSave
Entrada	parameters
Descripción	Genera una solicitud, llamando al ajax a conjunto de direcciones URL en la opción 'serverScriptSave'. El script guarda por defecto la imagen en el servidor, pero puedes modificarlo para hacer lo que quieras. Si el argumento es un objeto, los valores serán pasados como parámetros a la petición.

método	cropDownload
Entrada	parameters
Descripción	Llama a conjunto de direcciones URL en la opción 'serverScriptDownload'. Devuelve la cabecera de un archivo de descarga. Usted es libre para modificarlo. Si el argumento es un objeto, los valores serán pasados como parámetros a la URL.

método	addParameters
Entrada	parameters
Descripción	Cualquier parámetro añadido con esta función se presentará con la solicitud de guardar o descargar. Los parámetros se debe pasar como un objeto (parámetro: valor).

LIBRERÍA **progress.js** (*lib/progress.js*)

DESCRIPCION:

Las funciones de esta librería se usan para enseñar el proceso de carga de la imagen en porcentaje con una barra de estado.

Se usa cada vez cuando hacemos alguna operación de edición sobre una imagen y hay que recargar la.

USO:

Crear el div con el nombre “bar_window”. Luego hay que ubicarlo en la posición donde quiere que aparezca la barra de estado.

función	setup_bar
Descripción	Crea dibujo gráfico de la barra de estado, usando las variables globales bar_length , bar_height, bar_color, bar_background y otros.

función	show_bar
Descripción	Llamar después de haber hecho el setup_bar . Para enseñar la barra de estado en navegador.

función	progress_bar
Descripción	Llamar después de haber hecho el setup_bar y show_bar . Arranca la barra de estado, esta barra nos enseña cuanto se queda hasta la carga completa de las imágenes que están cargándose. Después se cierra sola.

función	close_bar
Descripción	Para cerrar la barra de estado.

función	cargando
Descripción	Para saber si la carga se ha completado o no.

4. ANÁLISIS DE RESULTADOS

Para realizar pruebas y conseguir los resultados más realísticos, nuestra aplicación final fue instalada en dos servidores. Uno se encuentra en mi red local de casa y otro está en el servidor de la Universidad Pública de Navarra.

La aplicación funciona con un navegador Firefox (v3.0, v3.5, v3.6) o Internet Explorer(v7, v8). En ambos navegadores todas las funcionalidades funcionan igual solo se difiere un poco en el aspecto de aplicación, pero esto se debe a diferencia entre los navegadores.

Cuando la implementación fue finalizada, se ha realizado la prueba completa del sistema con los datos reales. Este paso permite una comparación precisa de la salida del nuevo sistema con a que sabe que es salida correctamente procesada, así como una buena sensación de cómo serán manejados los datos reales.

Para probar el sistema se hicieron diferentes tipos de pruebas:

Pruebas de unidad: a medida que se hacían los módulos se probaba que funcionasen de forma individual correctamente. Durante esta fase se corrigieron muchos errores de todo tipo. Para poder corregir los errores fue activado el aviso de errores en el navegador.

Pruebas de diseño: una vez comprobado el correcto funcionamiento individualmente, se paso a conectar los módulos. Se corrigieron algunos errores.

Pruebas de validación y sistema: se corrigieron errores y se probó que el proyecto cumplía con todos los requisitos.

Fue medido el tiempo de descarga de página de Editor de imágenes. El tiempo de descarga de la página se refleja en la tabla:

Servidor	Editor sin imágenes (Tiempo en seg.)	Editor con imágenes de 1Mb en total (Tiempo en seg.)	Subir imagen de 300 Kb en Editor (Tiempo en seg.)
Universidad Pública de Navarra	10	12,1	25
En Red local	2,2	3	0,4

El tiempo de realización de algunas operaciones que dispone el editor con una imagen de 300 Kb(1531 X 2175), ver tabla de la pag.51:

Servidor	Rotar (Tiempo en seg.)	Recortar 100X100(Tiempo en seg.)	Redimensionar (Tiempo en seg.)
Universidad Pública de Navarra	X (La funcionalidad no habilitada a causa de S.O.)	2,5	3,8
En Red local	13	1	1,5

5. COSTE Y PLANIFICACIÓN

5.1 PLANIFICACIÓN

A continuación se detallara la organización y el tiempo del trabajo realizado para el análisis, diseño y la implementación.

La organización de trabajo se divide en dos partes:

Trabajo de analista: determinar los requisitos, estudiar el entorno de trabajo, elegir el software a usar y diseñar el sistema.

Trabajo de programador: codificación y pruebas.

Planificación de tiempo de proyecto es la figura 11.

Análisis	19 días
Análisis de los aplicaciones similares	12 días
Especificación de requisitos	7 días
Diseño	29 días
Diseño de interfaz graficas	15 días
Diseño de los diagramas	4 días
Diseño del modelo entidad relación	10 días
Implementación	83 días
Codificación	83 días
Pruebas	5 días
Realizar pruebas	5 días
Documentación	10 días
Documentación	10 días

Fig.11

El tiempo total de trabajo realizado es de 146 días (trabajando 3 horas/día), divididas en 88 días de trabajo de programador y 58 días de trabajo de analista.

5.2 COSTES

Se estima que un analista junior cobra 20 euros/hora y un programador junior cobra 15 euros/hora. El trabajo se realizaba aproximadamente 3 horas al día entonces:

Trabajo de	Horas	Euros/hora	Total a pagar en €
Analista	174	17	2958
Programador	264	13	3432
Total	438		6390

6. CONCLUSIONES Y PASOS FUTUROS

6.1 CONCLUSIONES

El presente proyecto, que consistía en hacer una aplicación empleando la tecnología Web 2.0, la cual permitiese a los usuarios el intercambio de sus trabajos de desarrollo o de espacios de trabajos de imágenes a los que vamos a llamar espacios de trabajos, aumentando de esta manera la creatividad y ofreciendo a los usuarios la posibilidad de acabar o ayudar en edición de imágenes de trabajos de los otros usuarios, fue cumplido satisfactoriamente.

Esta aplicación, nos proporciona una enorme facilidad de uso, gracias al empleo de la tecnología Web 2.0 para poder arrastrar las imágenes como se hace en el Windows.

La aplicación forma una parte de una página que es una galería de imágenes, y cualquier usuario que está observando las imágenes en alguna de las categorías tiene la posibilidad de editar las o subir sus propias imágenes para su posterior edición. También puede guardar los trabajos de edición que están en el desarrollo para ser seguidamente desarrollados por otros usuarios. Así que este objetivo también fue cumplido.

En nuestra aplicación el usuario o administrador puede elegir y buscar una categoría y también añadir cualquier imagen en la categoría que quiera. Por otra parte tiene la posibilidad de visualizar, añadir y editar todas las imágenes de que se dispone en cada categoría. Los usuarios disponen de una categoría de imágenes especial. Es de fotos privados, son las fotos subidas por un usuario y a los que no podrán ver otros usuarios. A su vez, también pueden realizar búsquedas básicas de las imágenes que les interesan. También pueden trabajar con los espacios de trabajo de los demás usuarios. Con los espacios de trabajo, los usuarios pueden visualizar, crear, editar, realizar búsquedas y guardar sus espacios de trabajo en el sitio de espacios de trabajo compartido. A los administradores les está permitido borrar cualquier imagen o espacio y a los usuarios solo imágenes o espacios de trabajo que están guardados por ellos.

La aplicación resultante permite editar las imágenes con operaciones como redimensionar, dibujar con diferentes colores, borrar, ampliar o reducir el espacio de trabajo(es cuando el menú con las operaciones y todas las imágenes se cambian de tamaño), rotar con cualquier ángulo, fusionar todas las imágenes, recortar, hacer el efecto de espejo, hacer copias de imágenes, crear hojas en blanco y guardar su trabajo en el sitio de espacios de trabajo compartido. Y todo esto también funciona correctamente con muchas imágenes a la vez moviendo las y situando las una encima de otra.

El editor resultante tiene un menú que se puede mover por cualquier lugar de espacio de edición. Las dimensiones del espacio de trabajo son muy grandes para así poder trabajar con las imágenes de cualquier tamaño. Cada funcionalidad del menú de edición tiene un dibujo asociado para que cualquier usuario que no sepa español pueda trabajar fácilmente con el editor. También tiene una barra de estado de carga para que el usuario sepa cuanto tiempo le resta para finalizar la descarga de una imagen después de cada operación. También en el editor el tiempo de espera fue reducido usando las tecnologías elegidas.

En conclusión todos los objetivos requeridos fueron cumplidos.

Para ver la diferencia entre el editor que hemos conseguido desarrollar y un editor online que fue elegido entre los 20 mejores editores online del mundo según la pagina web oficial <http://mashable.com/2008/11/13/online-image-editors-2/> , voy a analizar las funcionalidades que ofrece.

Para poder elegir un editor entre los 20 mejores, se tenía en cuenta que debe ser implementado con Ajax como el nuestro y no con Flash, porque con Flash es mucho más fácil hacer una aplicación, por eso los editores hechos con flash son mejores.

Y el editor elegido es: Pixenate.com, ver la figura 12.



Fig.12

Lo que nos importa es la diferencia entre las funcionalidades de estos dos editores, así que vamos a hacer una tabla con comparaciones:

	Pixenate.com	Imagix(nuestro editor)
Dibujar	Si	Si
Ampliar espacio de trabajo	Si	Si
Deshacer operaciones	Si	No
Rotar con cualquier ángulo	Si	Si
Recortar	Si	Si
Redimensionar	Si	Si
Voltear horizontalmente	Si	Si
Copiar Imagen	No	Si
Fusionar	No	Si
Edición simultánea de más de una imagen	No	Si
Cambiar el contraste, saturación.	Si	No

Normalizar imagen	Si	No
Cambio de ojos rojos	Si	No
Barrita mágica	No	No

No se puede decidir si uno es mejor que el otro, porque cada uno tiene su parte buena y su parte mala.

6.2 PASOS FUTUROS

Para empezar habría que completar nuestro editor con funcionalidad de deshacer las operaciones que importa mucho para un usuario. Esto se podría conseguir guardando todas las imágenes, cambiados con las operaciones anteriores, en una carpeta. Así vamos guardando todos los pasos de un usuario.

Como complementario de las funcionalidad de dibujar se podría hacer una nueva operación, es dibujar líneas, círculos o rectángulos. Es fácil de conseguir usando las mismas librerías que sirven para la funcionalidad de dibujar.

Muy interesante sería conseguir hacer una funcionalidad que permite a los usuario escribir un texto sobre la imagen. Para esta funcionalidad ya existen funciones que lo hacen en GD library, solo hace falta hacer que el usuario escriba el texto en el navegador y esto hay que pensar cómo hacerlo con javascript.

También sería útil tener implementada la barrita mágica que permite seleccionar una parte de imagen según un color (modificando el porcentaje de variación del color). Podríamos hacer con esta área seleccionada algo como llenar con un color cualquiera. Por lo tanto la funcionalidad de llenado con un color también habría que implementarla también. Pero para implementar la barrita mágica es necesario ser un experto en javascript. Además no he visto ningún editor online que tenga esta funcionalidad (que no sea programado con Flash).

7. MANUAL DE ADMINSTRADOR

Manual de instalación:

El sistema operativo puede ser cualquiera, solo si es posible realizar siguientes instalaciones:

1. Instalar Apache,PHP,MySQL,PhpAdmin
<http://www.maestrosdelweb.com/editorial/phpmysqlap/>
2. Instalar GD library para PHP
<http://www.php.net/manual/en/image.installation.php>
3. Instalar ImageMagick y imagick para php
<http://www.php.net/manual/en/imagick.setup.php>

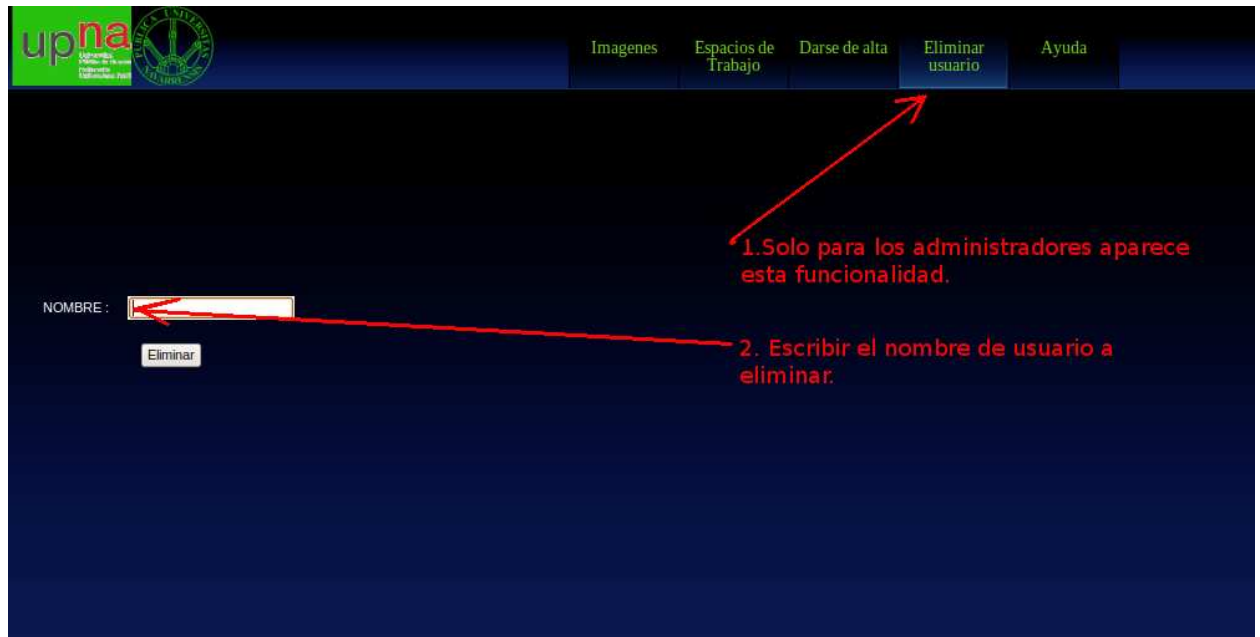
Crear la base de datos con el nombre imag_upna y con el usuario imag_upna, dándole todos permisos.

Insertar en esa la BD creada las siguientes tablas:

```
CREATE TABLE IF NOT EXISTS admines (  
  nombre varchar(255) NOT NULL,  
  fecha_in date NOT NULL,  
  PRIMARY KEY (nombre))  
CREATE TABLE IF NOT EXISTS codigo_categoria (  
  codigo int(4) NOT NULL,  
  nom_categoria varchar(255) NOT NULL,  
  PRIMARY KEY (codigo))  
CREATE TABLE IF NOT EXISTS constantes (  
  sec_orden_espacios int(15) NOT NULL)  
CREATE TABLE IF NOT EXISTS edit_datos_imagen (  
  nom_imag varchar(240) NOT NULL,  
  posX int(7) NOT NULL,  
  posY int(7) NOT NULL,  
  prop_user varchar(100) NOT NULL,  
  sizeX int(7) NOT NULL,  
  sizeY int(7) NOT NULL,  
  Ncapa int(10) NOT NULL,  
  angulo int(4) NOT NULL,  
  PRIMARY KEY (nom_imag,prop_user))  
CREATE TABLE IF NOT EXISTS espacios_trabajo (  
  fecha date NOT NULL,  
  nombre_espacio varchar(255) NOT NULL,  
  cod_dir_imag_ref int(12) NOT NULL,  
  nombre_user varchar(255) NOT NULL,  
  PRIMARY KEY (cod_dir_imag_ref))  
CREATE TABLE IF NOT EXISTS espacios_trabajo_imgs_info (  
  nom_imag varchar(240) NOT NULL,  
  posX int(7) NOT NULL,  
  posY int(7) NOT NULL,  
  codigo_ref_espacio int(12) NOT NULL,  
  sizeX int(7) NOT NULL,  
  sizeY int(7) NOT NULL,  
  Ncapa int(10) NOT NULL,  
  angulo int(4) NOT NULL,  
  PRIMARY KEY (nom_imag,codigo_ref_espacio))  
CREATE TABLE IF NOT EXISTS galeria_imag (  
  nombre varchar(255) NOT NULL,  
  orden int(9) NOT NULL,  
  categoria int(4) NOT NULL,  
  orden_todo int(15) NOT NULL,
```

```
insertadoBY varchar(255) NOT NULL,  
PRIMARY KEY (nombre))  
CREATE TABLE IF NOT EXISTS user (  
nombre varchar(200) NOT NULL,  
pass varchar(200) NOT NULL,  
PRIMARY KEY (nombre))
```

Eliminación de usuario.



Eliminación de cualquier imagen o espacio de trabajo.



8. MANUAL DE USUARIO

Eliminación de una imagen o un espacio de trabajo.



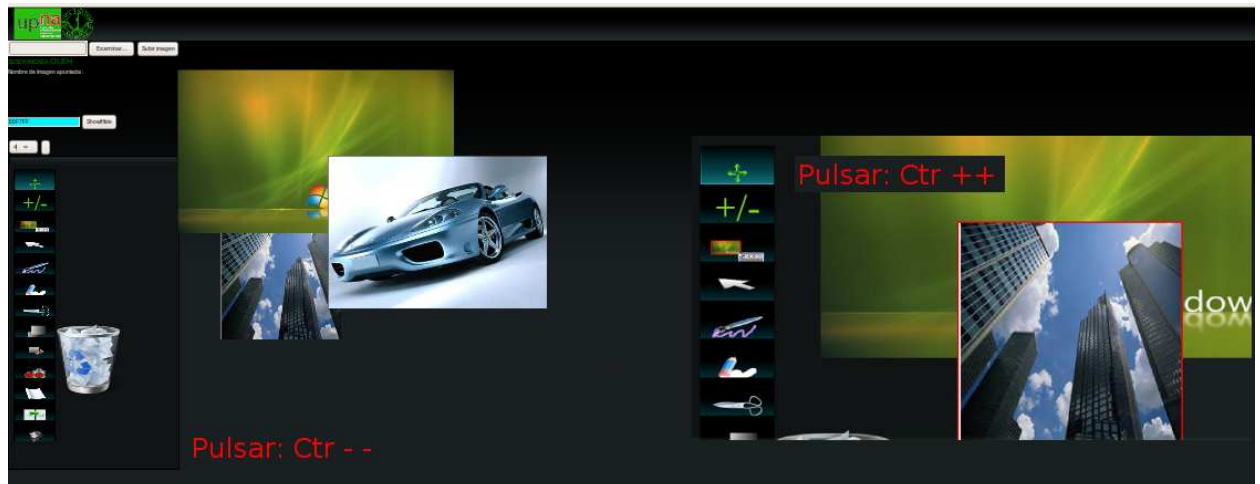
Subir una imagen a una categoría.



Lanzar editor.



Ampliar o Reducir el espacio de trabajo.



Subir o borrar una imagen de editor.



Redimensionar imagen.



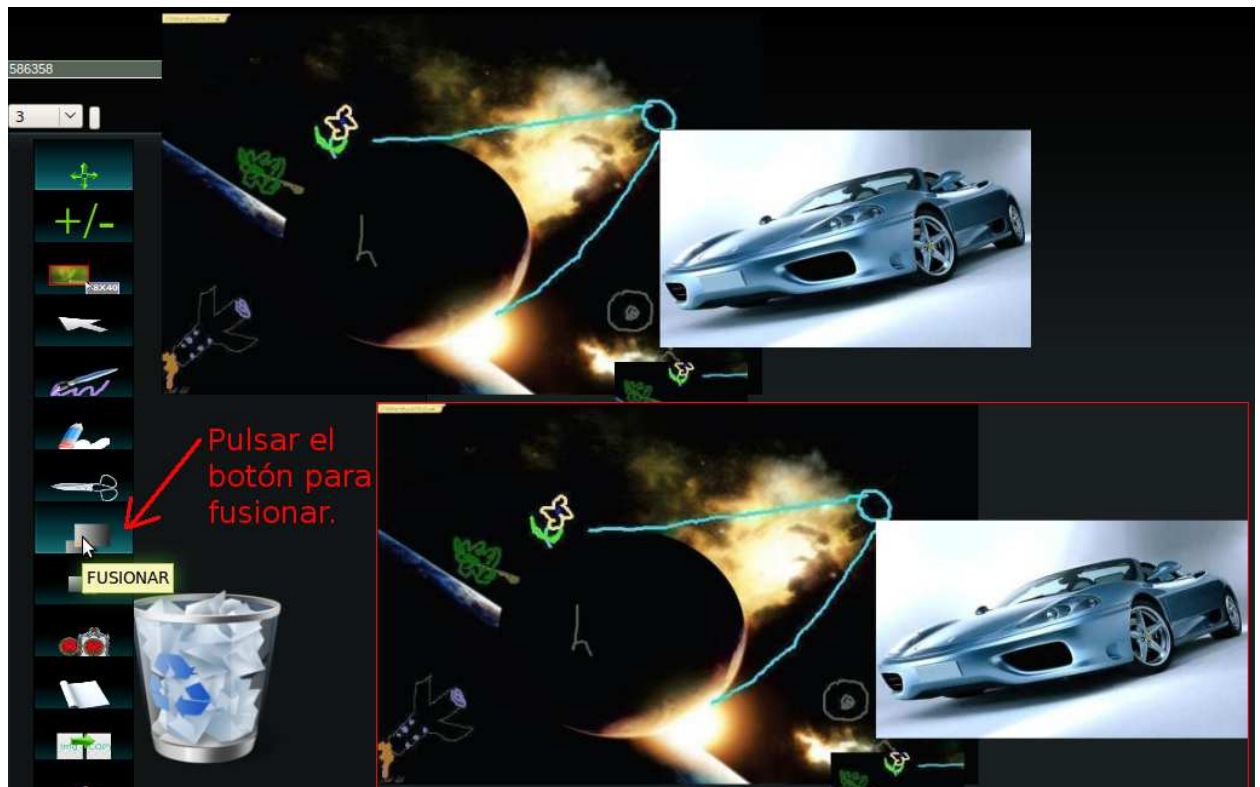
Dibujar o borrar sobre una imagen.



Recortar una imagen.



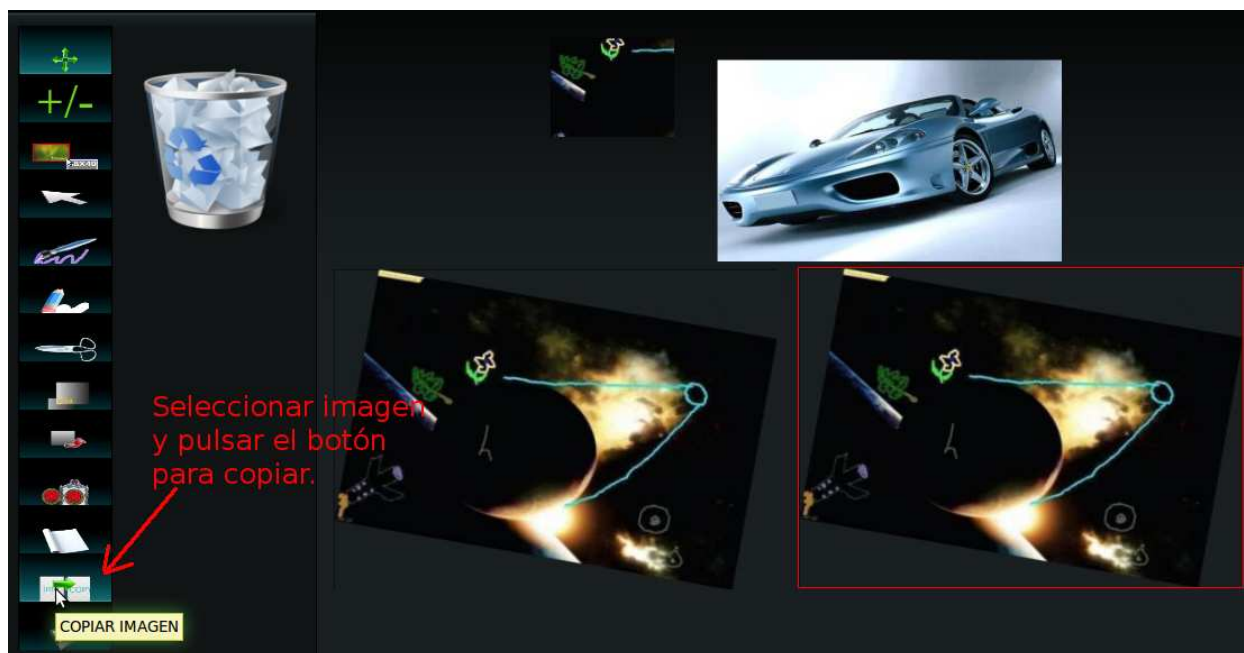
Fusionar imágenes.



Rotar imagen.



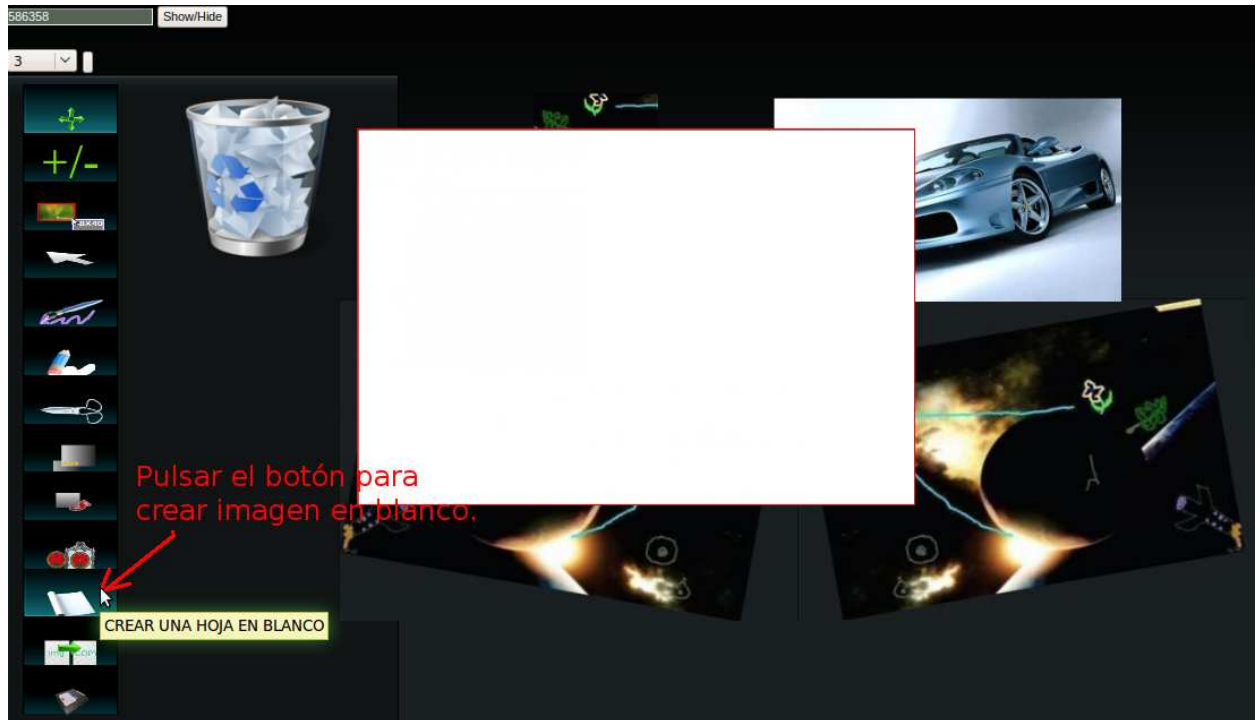
Copiar una imagen.



Voltear horizontalmente una imagen.



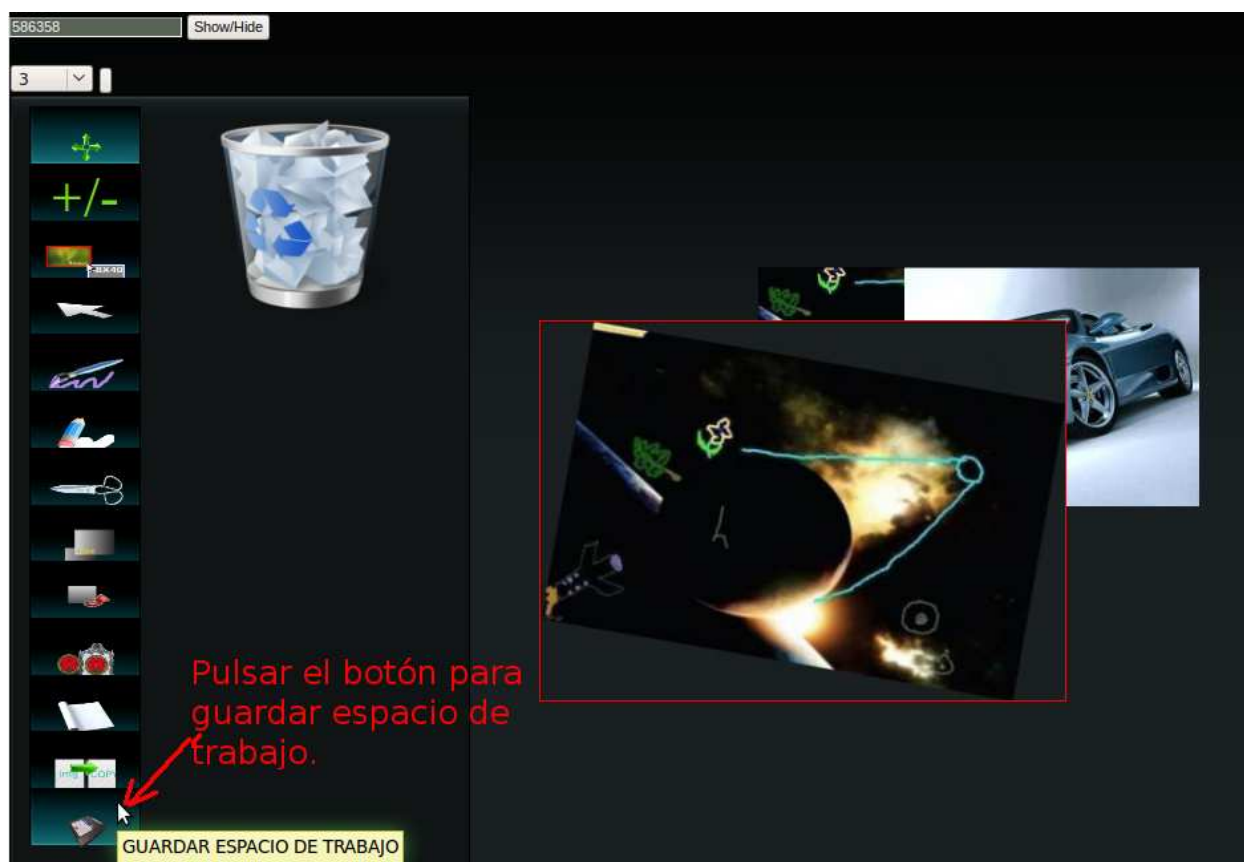
Crear imagen en blanco.



Guardar una imagen en su ordenador.



Guardar espacio de trabajo.



9. BIBLIOGRAFÍA

Todo sobre javascript:

<http://www.manualdejavascript.com/>

Todo sobre AJAX (Asynchronous JavaScript And XML):

<http://www.ajaxya.com.ar/>

Todo sobre PHP:

<http://www.desarrolloweb.com/php/>

Uso de GD library por php:

<http://www.php.net/manual/en/book.image.php>

Uso de ImageMagick por php:

<http://www.php.net/manual/en/book.imagick.php>

Instalar Apache,PHP,MySQL,PhpAdmin:

<http://www.maestrosdelweb.com/editorial/phpmysqlap/>

Instalar GD library para PHP:

<http://www.php.net/manual/en/image.installation.php>

Instalar ImageMagick y imagick para php:

<http://www.php.net/manual/en/imagick.setup.php>

Aplicaciones o sistemas similares:

Pixenate.com